GEOLOCATION OF RF EMITTERS USING A LOW-COST UAV-BASED APPROACH

THESIS

Michael A. Magers, Civilian, USAF

AFIT-ENY-MS-16-M-258

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-16-M-258

GEOLOCATION OF RF EMITTERS USING A LOW-COST UAV-BASED APPROACH

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Michael A. Magers

Civilian, USAF

March 2016

AFIT-ENY-16-M-258

GEOLOCATION OF RF EMITTERS USING A LOW-COST UAV-BASED APPROACH

Michael A Magers

Civilian, USAF

Committee Membership:

Dr. Richard G. Cobb
Chair

Dr. David R. Jacques
Member

Dr. Matthew J. Dillsaver
Member

Abstract

The proliferation of unmanned aerial vehicles (UAVs) in both military and civilian settings has prompted great interest in finding new and innovative ways to utilize these tools. One such application is to locate ground-based radio emitters from a UAV platform. The goal of this research is to study the feasibility of a low-cost (on the order of $1000) UAV geolocation platform. To accomplish this goal, a series of both real-world flight testing and computer simulated scenarios were conducted. Simulations for different sensor uncertainties and approach path scenarios such as loiter and button hook patterns were investigated. Results showed that a high uncertainty sensor of ±10 degrees was able to reliably geolocate the target provided it could fly sufficiently close to the emitter location. For the physical testing, a commercial-off-the-shelf Doppler direction finding unit was chosen as the method of performing the geolocation. Ground testing proved promising, locating the emitter to within 20 meters. However, flight testing showed poor results and was unable to locate the target. Areas of future work that could improve upon these results include investigating how altitude and antenna orientation variations caused by the movement of the aircraft affect the performance of the direction finding unit.

Acknowledgements

I would like to thank my advisor, Dr. Richard Cobb, and my supervisor, Mr. Jack Browne, for being patient with me and guiding me through this process. I would not have been successful in this endeavor without them. I would also like to thank the instructors of the small UAS design course here at AFIT, Dr. David Jacques and Dr. John Colombi, for providing guidance and expertise during classroom portion of this project. I would also like to thank the other students in the small UAS design course: Namkyu Kim, Kyle Hathaway, Jason Png, Pat Grandsaert, Mitch Gillespie, and Stefan Hardy. This project would have never come to fruition without the hard work of all of the team members. Lastly, I would like to thank my friends and family for believing in me and providing emotional support throughout the project.

Michael A Magers

Table of Contents

Page

List of Figures

Page

List of Tables

Page

GEOLOCATION OF RF EMITTERS

USING A LOW-COST UAV-BASED APPROACH

I. Introduction

1.1 Overview

The use of unmanned aerial vehicles (UAVs) is becoming more and more prevalent in the Air Force of today. These UAVs provide many benefits over traditional manned aircraft. In general they are cheaper to develop, cheaper to build, and cheaper to operate. They are often more efficient and can remain on station for tens of hours at a time which would be impossible with a manned aircraft. One of the largest benefits is that they are able to keep human lives safe and away from hostile airspace. For this reason, UAVs are able to go into areas that a manned aircraft would not want to go. UAVs are also about to be used in the national airspace, opening up even more applications and furthering the proliferation of these platforms. For these reasons, applications and advancements of UAV technology are of high interest to the Air Force.

Another area of rapidly growing interest is that of geolocation of radio frequency (RF) emitters. Many devices in today's highly connected and electronic world emit some sort of radio signal. Example devices include cell phones, walkie-talkies, and wireless routers. A geolocation package provides the utility to locate the area from which these signals are coming. This provides many possible applications such as locating a hiker lost in the woods or a friendly troop separated from their unit, to finding a hidden enemy base of operations.

This research looks at combining an RF geolocation package with a UAV to form an aerial platform capable of finding a hidden emitter located on the ground. This topic currently has much room for advancement and can provide real, tangible benefits to the Air Force.

1.2 Problem Statement

As UAVs become more prevalent in the battlespace abroad and friendly airspace at home, more applications are constantly sought on how to use these systems in new and creative ways that can bring a variety of benefits. The specific problem and source of investigation that this research deals with is that of performing geolocation with these aerial platforms. In particular, this research attempts to help answer the question of how effective low-cost systems, on the order of $1000, are in performing the task of geolocation, and to what degree of success they are capable. Upon answering these questions, the research also aims to provide suggestions on how best to go about performing the geolocation process, and to provide recommendations on ways to improve upon the work in the future.

1.3 Research Objectives/Questions/Hypotheses

The objective of this research is to investigate and provide further proof of concept for the practice of locating a ground-based RF emitter from an unmanned aerial platform. This topic is currently of high interest due to the wide range of potential applications, however the scope of the current flight demonstrations with low-cost hardware are limited. This research aims to contribute more real-world data and experience on the technique of performing geolocation from a low-cost aerial platform.

1.4 Research Focus

The focus of the research presented in this thesis is to investigate and test how well a low-cost UAV system can geolocate a ground-based radio target. The research is split into both real-world flight testing as well as computer simulations. The focus is to determine if a low-cost airborne platform is capable of locating an RF emitter on the ground and to what degree of success it is capable of doing so, and how these results can be improved upon. Assumptions and limitations that help shape the scope of the work and provide a framework for results are detailed in Section 1.7.

1.5 Investigative Questions

This research seeks to answer the following three specific questions: 1) What is the achievable accuracy of a low-cost system in locating a ground-based emitter, 2) What sort of search pattern is best utilized for locating the target, and 3) Given more time for investigation, what additional work could be done to improve the performance of the system.

1.6 Methodology

This research begins with investigating the current methods available for performing geolocation of an RF emitter. These methods are discussed in detail in Chapter II. Chapter II also discusses products that are made to perform geolocation and analyzes their suitability for use on an unmanned platform. These devices include things like Doppler antenna systems and bow-tie antenna systems.

After the basic methods are examined, the focus then shifts towards attempting to determine which approach is most suitable to address the main problem of the research, which is geolocating an RF emitter with a low-cost UAV package. Many challenges are present when attempting to adapt a geolocation system to an airborne platform, particularly a small, lightweight UAV. These include being out of the plane (elevation) of the emitter, and the constantly moving orientation of the geolocation system itself in relation to the emitter. A series of post-processing corrections can be made to try and alleviate some of these challenges, and are discussed in detail in Chapter III.

In addition to using flight test to determine the feasibility of the geolocation system, a series of simulations were also performed. These simulations seek to complement the flight test and attempt to model some of the results that were observed. The results seen during flight test drove some of the decision making process behind how to construct the simulations. Key among these decisions was how to model the accuracy of the geolocation being performed. Overall, the goal of the simulation work was to provide further feasibility analysis of the airborne geolocation problem, and to make recommendations on which maneuvers are more efficient at performing geolocation compared to others. The full discussion of the methodology behind the simulations is found in Chapter III.

1.7 Assumptions/Limitations

Some limiting assumptions have been made to make the problem manageable within the constraints of available time and equipment. It is assumed that the emitter is located on the ground, that it is stationary, and that it is somewhere within the specified

target region of interest. It is also assumed that the frequency that the signal is being emitted at is known and unchanging. Another assumption is that the signal broadcasts continuously. It is also assumed that the emitter is located in an area free of other signals or large structures that could cause noise, interference, or multi-path effects. Lastly, for the purpose of performing the simulation analysis, the exact method of performing the geolocation is disregarded; it is simply assumed that performing the geolocation is possible. Further rationale for assumptions made in the analysis can be found in the simulation methodology section of Chapter III.

The main limiting factor that shaped the scope of the work was cost. There are currently functioning UAV-based geolocation systems that are able to perform the tasks discussed in this research. However, those systems are much more sophisticated and costly, typically in the tens to hundreds of thousands of dollars. That degree of expenditure was outside the interest area of this study. Instead, this research is limited to systems in the hundreds to low thousands of dollars range which would be suitable as expendable devices in many applications. A more specific equipment budget is discussed in Chapter III.

1.8 Implications

The results of the research detailed in this thesis are projected to have many beneficial applications. First, this research will help move low-cost aerial-based geolocation systems one step closer to becoming a reality. By making the package airborne, many benefits are realized over the traditional ground-based systems that are currently in use. These benefits include being able to more quickly search a large target

area, and more easily cover adverse terrain such as swamps, bodies of water, and mountainous areas that would pose problems for ground-based systems. An additional benefit is the reduction in danger to human life since these airborne systems are able to be piloted remotely, away from the hazardous zones that might require searching.

Second, this research will help the military move one step closer to the goal of having a low-cost, aerial-based, passive geolocation package. The U.S. military would have numerous possible uses for such a system. These possible uses include locating enemy whereabouts based on radio communication without detection since a passive system requires no interrogation of the source emitter, finding friendly troops that may be lost but able to transmit on radio, and possibly locating low power emitters such as those found in improvised explosive devices (IEDs). All of these possibilities would increase situational awareness of the warfighter and decrease the risk and exposure to dangerous situations. In addition to the above benefits, the low-cost aspect of the research is also of great interest to the military. Low-cost and easy to implement systems are of paramount importance especially in times of reduced budgets when agencies are looking to stretch funding as far as possible. This research will focus on systems that are on the order of a few thousand dollars to build and operate, leading to a large amount of possible savings over systems that are hundreds of thousands of dollars.

Lastly, this work can also be used and expanded upon by the small UAS design sequence here at the Air Force Institute of Technology (AFIT), through which this research was started. Additional work could build upon the preliminary work presented in this thesis and lead to further refinement of the system and a better overall solution.

1.9 Preview

This research is broken down into five chapters. Chapter I provided the general problem that is to be addressed along with motivation for the work and goals for the outcome. Chapter II provides background for the work and discusses the current state of the research pertaining to performing geolocation from a UAV. Chapter III discusses the methodology used for obtaining the results of the work. This methodology includes the approach taken for both simulated and real-world results that were obtained. Chapter IV analyzes the results that were achieved from the both the simulated scenarios and the real-world flight testing. Lastly, Chapter V provides conclusions of the work and suggestions for follow on research that could be done to further the results and improve upon the findings.

II. Literature Review

2.1 Chapter Overview

The purpose of this chapter is to provide a background for the relevant material relating to geolocation technology. This includes both general geolocation theory as well as discussion of specific devices that perform geolocation. This chapter also provides an overview of the current state of the research pertaining specifically to geolocation performed from a UAV platform. Reviewing the current state of the research will provide a framework for what has been accomplished up to this point and to point out areas of interest in which future work can still be done to further the technologic capabilities of UAV-based geolocation. The goal is to give a sense of what has been accomplished so far, what gaps still exist, what improvements can be made, and give context for where this research fits in relation to the current body of work.

2.2 Geolocation Methods

This section reviews current literature on the methods and techniques of locating ground-based radio frequency emitters and their usage on unmanned aerial vehicles. This topic is of increasing interest to both the military and civilian world as UAVs become both cheaper and more prominent in the air space. The specific topic of geolocation is motivated by the small UAV design sequence offered at the Air Force Institute of Technology, through which this research was started. The mission problem for the sequence was to produce a functioning geolocation system comprised of a geolocation package affixed upon an airborne platform and operated by an on-site ground station.

Further motivation comes from the many possible applications of such a geolocation system, including locating lost hikers in possession of a radio or locating an enemy's mobile command center. The goal of this section is to provide an overview of different geolocation strategies and their applications to airborne platforms and also to identify areas where more research can be performed.

2.2.1 Geolocation Overview.

The term geolocation simply means to determine the location of something on the Earth, usually specified in some given reference frame. In the context of this work, *direction finding* is the process of using receivers at known positions to produce a range or direction measurement from the receiver to the target, and *geolocation* is the process of combining these measurements and calculating a specific target location estimate. By exploiting the physics of a wave propagating from its source and combining that information with the knowledge of where the receivers are located, the determination of the target location can be derived in many different ways.

Many different methods have been developed for use in geolocating an RF signal and many products that perform geolocation are commercially available. Five main methods of geolocation will be outlined: 1) the angle of arrival (AOA) method which locates the signal position from the directional angle of the signal, 2) the frequency difference of arrival (FDOA) method which locates the signal position from the difference in frequency of the signal measured between two receivers, 3) the time of arrival (TOA) method which locates the signal from the precise time the signal arrives at multiple receivers, 4) the time difference of arrival (TDOA) method which locates the

signal position from the time difference in detection of the signal from multiple receivers, and 5) the received signal strength (RSS) method which locates the signal based on received signal strength. These five methods will now be explained in more detail.

2.2.2 Angle of Arrival (AOA).

Geolocation by the AOA method is done in a two-step process. The first is to estimate the angle at which the received signal is located in relation to the receiver. This is typically done using a phased array antenna which consists of an array of sensors and a signal processing device [1]. As a signal is received, the time or frequency difference detected at different receivers can be used to make an estimation of the angle at which the signal came from. As the receivers detect the signal, an angle estimate is generated from each receiver to the signal. This then leads to the second step of the process. The angle estimations are used to generate lines of bearing (LOBs) from the known location of the sensor to the estimated target position. The emitter location is then determined from calculating the intersection of these LOBs. A diagram of this general calculation is shown in Figure 2.1. The AOA method differs from other approaches detailed in this chapter by the fact that the information given is in the form of bearings relative to the receiver, as opposed to range information that is given by the other methods.

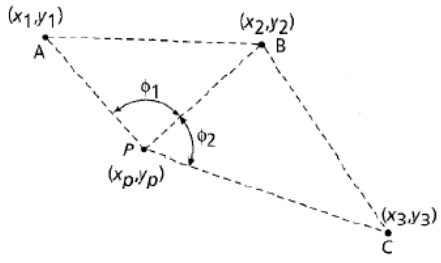


Figure 2.1: Geolocation through a LOB-based approach [2]

There are multiple benefits of using an AOA approach. Only two receivers are needed to perform the geolocation instead of three, since two lines of bearing will produce a single unique intersection point in 2-D space. However, since in practice bearing measurements are experimentally observed quantities that are subject to noise and error, they will generally not pass through the true target position [3]. Another benefit is that no precise time synchronization amongst receivers is needed like it is in other methods. The drawbacks to the AOA method include the need for complex, expensive smart antennas that are capable of performing the AOA calculations, as well as the need for line of sight (LOS) to the target [4].

2.2.3 Frequency Difference of Arrival (FDOA).

If either the target emitter or one of the receivers is moving, a phase shift in the signal will occur [5]. This results in each receiver detecting a slightly different frequency that can then be used to estimate the position of the target. Curves of possible locations for the emitter can be generated by knowing the location of the observation points, the velocity of the receivers, and the amount of observed phase shift. This is depicted in Figure 2.2. After multiple curves are generated by taking measurements from different locations, the resulting intersection point is the estimated location of the emitter.

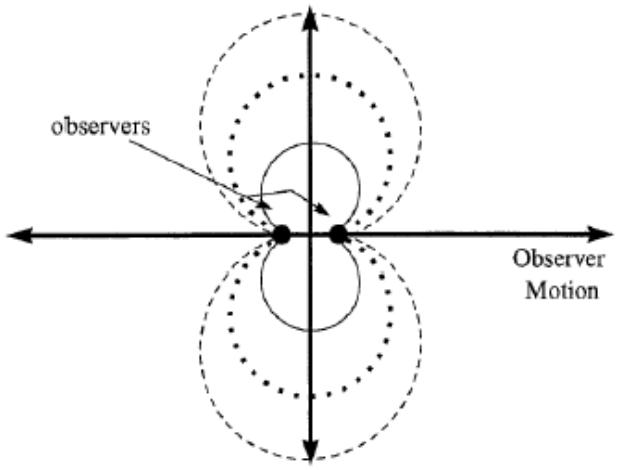


Figure 2.2: Geolocation based on FDOA contour lines [5]

One benefit of the FDOA technique is that it produces emitter location error estimates that are often in a different direction than that given by the more widespread TDOA method [6]. This allows for use of a hybrid method between the two systems that can give added accuracy over a single method. This is shown in Figure 2.3.

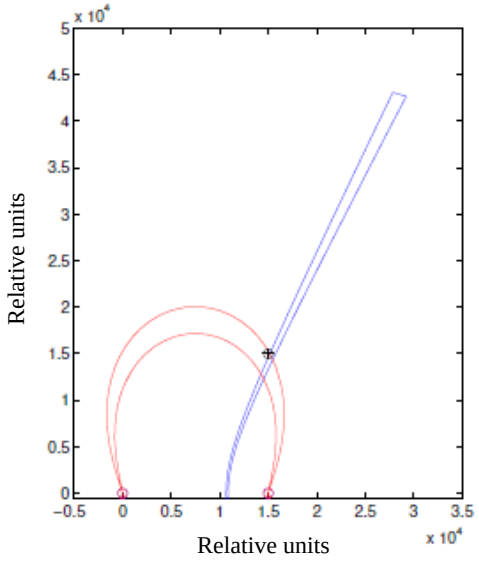


Figure 2.3: FDOA measurements (red) combined with TDOA measurements (blue) [7]

The drawback to the FDOA method is that it is difficult to implement on low-power systems. The receiver velocity must be sufficiently high to produce a large enough shift to overcome the error in measuring the frequency, and the accuracy of the measurements must be better than the smallest expected frequency shift in order to detect it above the noise present in the system.

2.2.4 Time of Arrival (TOA).

The TOA method of geolocation is based on the principle that waves propagate at the speed of light, which is a constant. The TOA method relies on being able to accurately determine the exact time that a receiver detected a signal as well as knowing exactly when that signal originated. By synching the time of arrival of the signal at the receiver with the time when the signal left the emitter, a distance can be calculated since the travel speed of the wave is known [5]. Since multiple points are located the same distance away from the receiver in space, a circle of possible locations is formed around the receiver. By taking measurements at different locations and finding the intersection point of these circles, the target position can be determined. This is shown in Figure 2.4.

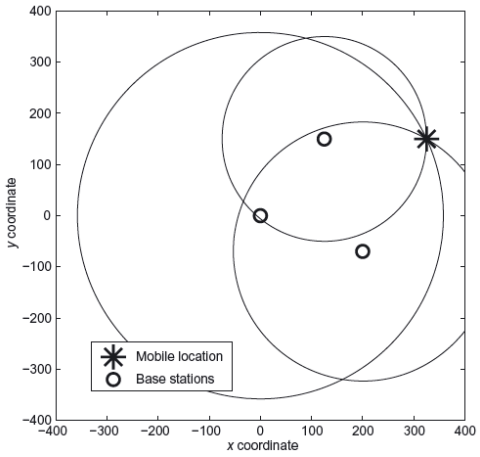


Figure 2.4: Range-based geolocation estimate based on TOA [8]

The benefit of the TOA technique is that it produces highly accurate geolocation results. The drawback to this technique is that unless it is used in a laboratory setting, it is rarely known exactly when a signal originated from an emitter. This severely limits the usefulness of this technique in real-world applications. This is where the time difference of arrival method comes into play.

2.2.5 Time Difference of Arrival (TDOA).

The TDOA method is similar to the TOA method except that the emitter does not need to be synched with the receivers. This method instead uses synchronized receivers and the time difference between the detection of a signal by multiple receivers to generate curves of possible positions. Each calculated time difference will result in a hyperbolic curve of possible target locations [9]. By taking multiple measurements and computing multiple curves, the resulting intersection of these curves will be the estimated target location. This can be seen in Figure 2.5.

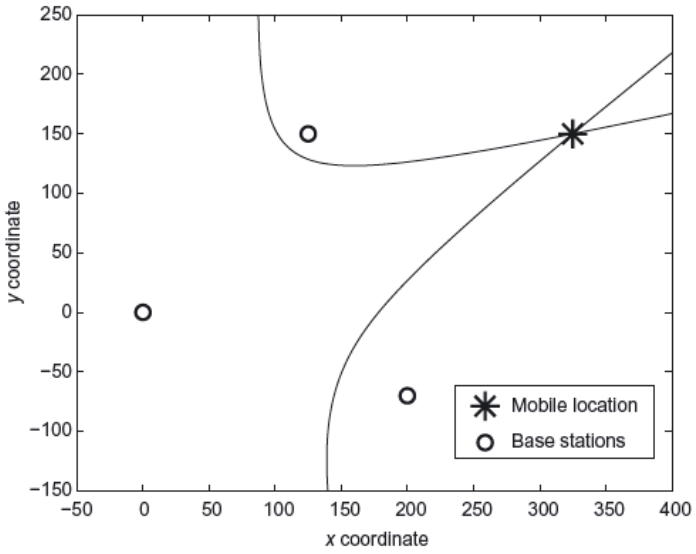


Figure 2.5: TDOA hyperbola position estimation [8]

The drawback to this technique is that a minimum of three receivers, which in this context means three UAVs, are required to instantly generate a target location if it is assumed to be on the surface of the Earth, and four receivers are required if the target altitude is unknown [10]. This is because two receivers are needed to generate one hyperbolic curve and a minimum of two curves are needed to locate a target in a 2-dimensional plane. This constraint can be overcome by moving the receivers and taking TDOA measurements at different locations, but this results in a non-instantaneous measurement, thus requiring more time for the geolocation to be performed and presenting further problems if the target is moving. Another drawback to this method is that the multiple receivers need to be precisely synched with each other for the geolocation to work. This can present problems for multiple UAVs where there are transmission time and bandwidth limitations between them.

2.2.6 Received Signal Strength (RSS).

The RSS method of geolocation works by measuring the strength of the received signal. As the receiver moves and the received signal strength grows stronger, the receiver is moving towards the location of the emitter. If the received signal strength declines, the receiver is moving away from the emitter. An example of increasing received signal strength is shown in Figure 2.6.

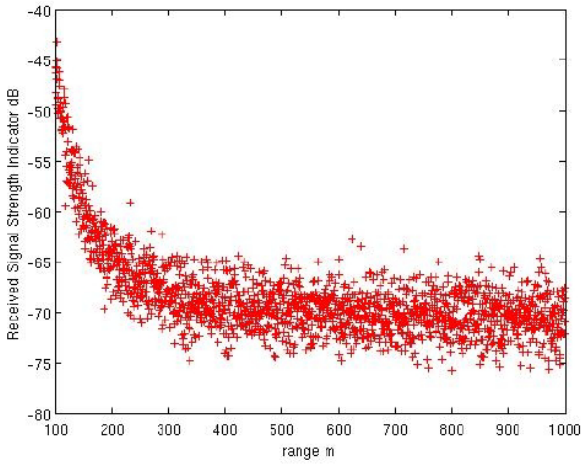


Figure 2.6: Received signal strength at different distances from emitter [11]

Some benefits of the RSS method of geolocation is that it is the most simple and easy to implement [12]. It is also readily available in almost any wireless infrastructure, and can often be combined with more sophisticated systems as a way of creating a hybrid approach that can improve accuracy. However, this method generally produces the least accurate position solution of the discussed geolocation methods and also requires the receiver to move to within a close proximity of the emitter due to the fact that the received power follows an exponential decay and that increased power levels may not be detected until moving very close to the target [13]. This can present a problem if the emitter is located in hostile airspace.

2.3 Direction Finding Devices

The first step in geolocating an RF emitter is to gather information being broadcast from that signal, and then estimate either the range or direction of that emitter relative to the receiver. From there, once a series of these estimates are made, they can be

combined to perform the actual calculation of the target's estimated position. This section outlines the theory of some commercially available devices that perform the direction finding aspect of geolocation.

2.3.1 Dipole Antenna Direction Finding.

A dipole antenna is the simplest and most common type of antenna. It consists of two identical conductive elements, usually metal rods or wires, with a feedline connected to each. They function by measuring the frequency of the incoming signal across the two elements. The most common type of dipole antenna is the half-wavelength antenna. In this set up, the two conductive elements are spaced so that the total distance between them is ½ of the total wavelength of the received signal. By doing this, the measured amplitude is at a maximum. By rotating the antenna to where a null is, that will give the direction that the signal emitter is in. This is because the measured wavelength difference across the two conductors at that point will be zero [14].

The nice thing about this set-up is that it is very simple and easy to implement. The downside is that the receiver must be specifically made for one signal, and if the signal is unknown or if it is changing then the usefulness of the antenna is degraded. Some of this may be mitigated by using a bow-tie antenna.

2.3.2 Bow-Tie Antenna Direction Finding.

A bow-tie antenna functions a little differently than a dipole antenna in that it functions based off angles instead of lengths, meaning that it can be more widely applicable to changing emitter wavelengths. By basing the measurement off of the angle of the received signal instead of the length, the bow-tie antenna has a much higher

bandwidth than a normal dipole antenna [15]. The downside is that the accuracy of the direction finding measurements is still not very high. For this reason, another method for direction finding is often used, the Doppler direction finding method.

2.3.3 Doppler Direction Finding.

One of the most common devices used in direction finding is a Doppler direction finder. This technique is popular due to the fact that it is still easy to implement but produces a higher fidelity solution than those given from the methods described above. The technique at work in these devices is what is actually called a pseudo-Doppler phenomenon. A traditional Doppler shift works through observing the change in frequency of a signal based on the motion of the emitter or receiver. As the two are brought closer a higher frequency is observed, while when the two are moving apart a lower frequency is detected. This can then be used to determine the direction from the receiver to the emitter. A pseudo-Doppler system works on the same principle, but does so without requiring movement of the emitter or receiver. This is done by quickly sampling through the dipole antenna elements of a circular array, like the one shown in Figure 2.7. By quickly measuring the frequency at different locations, the Doppler effect is simulated without any motion being required [16]. This allows these systems to function even when both the receiver array and emitter are stationary.

An additional benefit of Doppler direction finding systems is that their combination of relative simplicity, low cost, and small footprint combined with a good degree of accuracy make them a prime candidate to be utilized in a UAV geolocation package. However, the integration effects between the antenna array and the airframe still

need to be investigated further. This leads to the next topic of discussion that shifts the focus away from general geolocation theory to the more specific application of UAV geolocation.

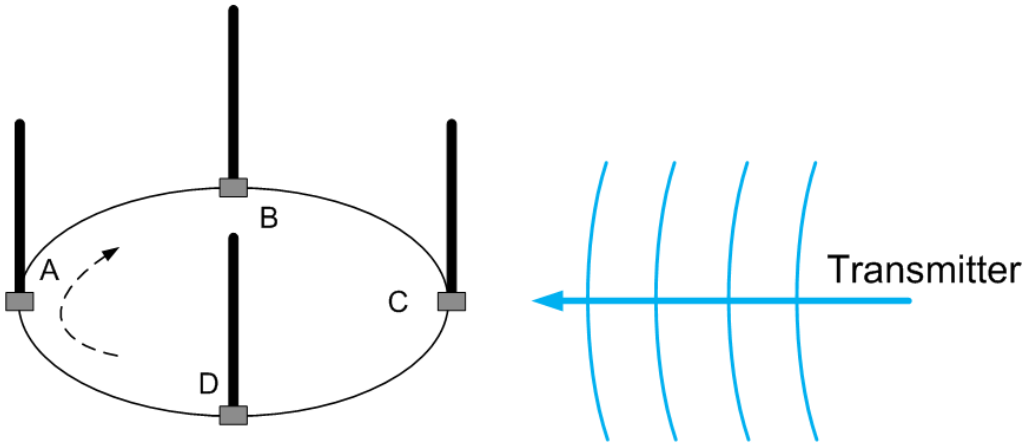


Figure 2.7: Antenna array set-up for Doppler direction finding [17]

2.4 UAV Geolocation

Now that the background of the common geolocation techniques has been covered, the specific topic of UAV geolocation will be examined. This survey of the literature will be broken into two main parts: simulated work and experimental work. The simulated work will focus on studies that have been performed researching the viability of UAV-based geolocation as well as how to locate a target efficiently under a variety of circumstances such as different numbers of targets or receivers, and different ways of performing the actual geolocation. The physical work portion will focus on testing that has been done examining the real-world applicability and success of UAV-based geolocation. The focus will be on relatively cheap and easy to implement systems,

generally characterized by low power and low computational capabilities, that would be similar to the flight testing done that is described in Chapters III and IV.

2.5 Simulated UAV Geolocation

Much of the work that has been performed studying the idea of geolocating a ground target from a UAV has come in the form of computer simulations. These simulations cover a wide variety of parameters, situations, and intended goals. This section will explore breadth of simulations that have been performed and attempt to give an accurate representation of where the current body of work is at in regard to studying the theoretical side of UAV-based geolocation. The discussion begins with the simplest case of a single UAV working to locate a single emitter, and then progresses to the more complicated situations of multiple UAVs and multiple emitters, as well as optimal route planning.

2.5.1 Single UAV Geolocation.

The concept of locating an emitter through the use of a single detection platform has been studied extensively. One of the first instances was by Stansfield in 1947 [3], although it should be noted that he used three receiver stations, but since nothing about the set-up was moving and time was of no concern, this functions the same as having one device taking measurements at three locations. He attempted to define what the most likely position on an emitter was based on three bearing measurements with a small degree of error, more specifically with a normal distribution with a standard deviation of 2 degrees. The introduced error is a necessity in all simulations due to the fact that without it the geolocation process is trivial, it can simply be determined from the

intersection of two perfect bearing measurements. Stansfield made some important observations. He observed that the uncertainty in position of the emitter increases rapidly with increased distance from the receivers, that the reliability of the location estimation did not depend on the size of the particular overlapping region from which it was derived assuming the error in the system was unchanged between trials, and that the odds were 3 to 1 that the true position of an emitter was outside the triangle formed by only three lines of bearing. An example of the triangle area formation from his work is shown in Figure 2.8, which shows LOBs generated from points K, L, and M for different emitter locations, with the most likely emitter location being marked by a dot in each case.

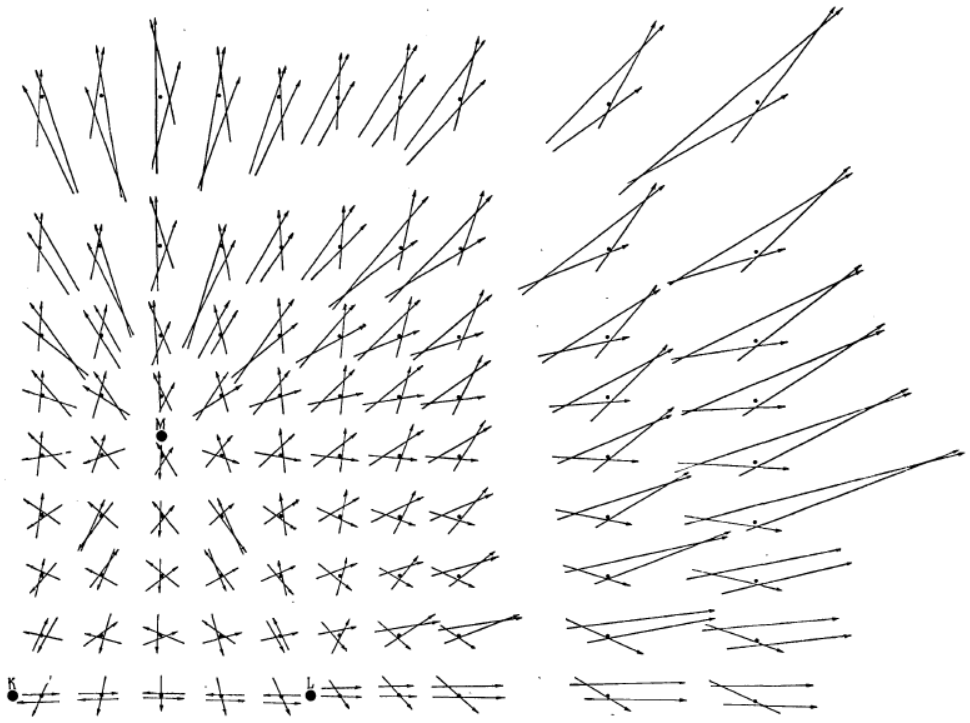


Figure 2.8: Overlap region from three LOBs showing most likely emitter location [3]

This early work showed that location of an emitter can be estimated from a small number of measurements, only three, but that the accuracy of that location was highly

dependent on the distance between the emitter and receiver, as well as the inherent error of the direction finding system.

From then on, the sophistication and number of variables being examined in single-target, single-platform geolocation has only grown. Some examples of these studies can be seen in [18-20]. Most studies are unconcerned with how the actual direction finding process is performed, they merely assume that is it possible. These studies mainly focus on the geometry of the geolocation once the lines of bearing from the UAV to the emitter are formed. One such study is [18] which proposes a method of geolocation based on computing the centroid of a continually updating polygon formed by the uncertainty in the bearing measurements taken from a moving UAV. This approach is very similar to the geometry shown in Figure 2.9. The study claims increased accuracy over a more standard Kalman filtering approach and also states the accuracy can be improved based on the selection of a proper flight path, although only a straight line and curved path were examined.

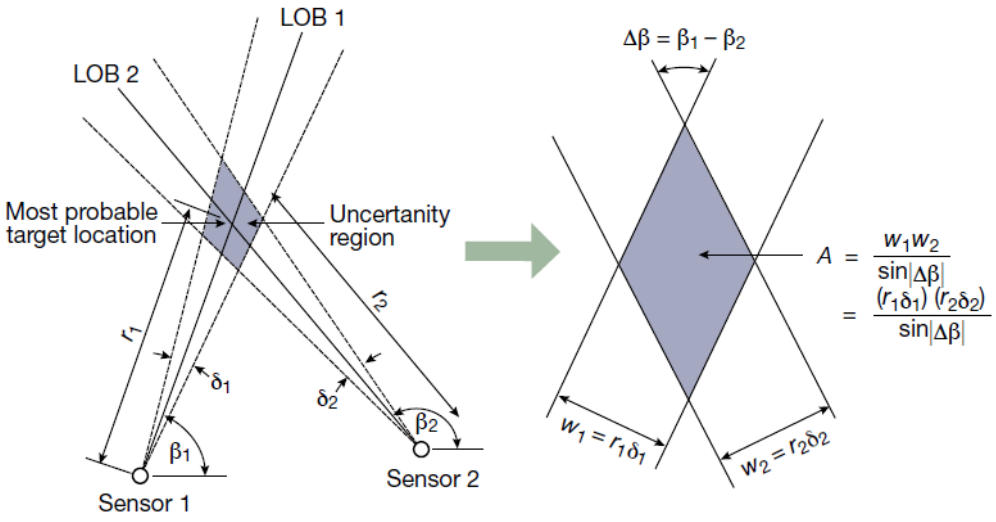


Figure 2.9: Line of bearing geolocation with uncertainty [21]

The overall conclusions from the single-emitter, single-UAV studies are that geolocation with one UAV is both possible and feasibly very accurate. The findings indicate that the overall accuracy is highly dependent on the uncertainty in the direction finding process and the location of the observation points relative to the emitter location. The trend is that less uncertainty in the direction finding leads to more accurate results, as does flying closer to the emitter location and taking measurements from a closer range.

The benefit of performing geolocation with a single UAV is decreased cost since fewer platforms are needed, increased simplicity since teams of UAVs do not need to be synchronized and working together. In addition, there is theoretically no loss in accuracy of the geolocation either. This is because a single UAV can take all the same measurements as a team of UAVs working together, and therefore arrive at the same solution. However this leads to the drawback of only operating a single platform, and that is time. An instantaneous solution cannot be formed by a single UAV since it needs to take measurements at different locations, and therefore has to take the time to physically move between those locations. A team of spatially separated UAVs can perform the same task instantaneously. The lack of an instantaneous solution provided by only a single UAV would become even more apparent in the case of a moving emitter. By having to move to acquire bearing measurements for geolocation, a UAV operating independently would constantly be working with outdated information caused by the movement of the emitter. A team of UAVs working cooperatively would alleviate this problem. Multiple UAV-based geolocation will now be discussed in the next section.

2.5.2 Multiple UAV Geolocation.

Much of the new research into the field of UAV geolocation has been focused on teams of UAVs working cooperatively to locate a target. The benefit of a team based approach is that target localization can be performed more quickly due to the increased number of receivers, meaning more measurements can be taken more quickly. The reason for increased focus on this subject is the increased computational power and bandwidth available to utilize on the platforms. This increase has allowed a level of synchronization and sharing amongst platforms that was impossible in the past. However, even modern systems can run into bandwidth and complexity issues using three or more UAVs [22]. As with the single UAV case, many different studies have been performed looking at a wide variety of aspects related to multi-UAV geolocation. Some examples of studies include [23-25]. Some of the main issues of research are examining the effects of different numbers of UAVs [11], more than one emitter [26], and effect of path planning [10]. While the issues of bandwidth and communication between UAVs is still a large hurdle in real-world applications, much of the literature does not examine the nuances of these effects, it is largely left as a hardware issue to be figured out.

A number of important conclusions can be drawn from the research that has been accomplished up to this point. One is that adding additional UAVs improves the speed and accuracy of the localization due to the increase in number of measurements. However, in addition to the obvious technical drawbacks of adding more and more platforms, the problem of diminishing returns becomes apparent after more than four UAVs are used [21].

Another important conclusion is that geolocation of both moving emitters and multiple emitters is possible [26,27]. The main conclusion of these studies is that performance of the localization is heavily dependent on the flightpaths of the UAVs, and that the concept of route planning is still being developed. This includes figuring out the actual logic of where to fly, as well as how that decision making would be performed in the real world. Both [9] and [28] discuss the ideas of route determination occurring either onboard the UAV or on some central processing system on the ground. The benefits of each are the drawbacks of the other. Calculations performed onboard the UAV allow decisions to be made quicker, since data does not need to be sent out to the ground station and then waited on for directions to be sent back. However, processing power onboard the UAV is limited and may not be able to handle optimal path planning calculations. The reverse is true for performing decision making on the ground. The next section will look at some of these aspects of route planning.

2.5.3 Route Planning and Optimization.

One of the most active areas of research going on currently in the field of UAV geolocation is that relating to route planning and optimal flight path determination. Since most UAVs currently fly preset routes or loiter patterns, moving to an efficient form of path planning has the potential to decrease geolocation times and increase geolocation accuracy, which could lead to increased mission performance.

Most path planning algorithms function in broadly the same way. A cost function is chosen that defines the parameters to be optimized upon, and from there a control is determined to perform that optimization. The most common parameter to optimize on is

the size of the error ellipse associated with the geolocation uncertainty [28], and the most common control is vehicle position, generally manipulated by heading angle due to the two-dimensional nature of most of the studies, and constrained by vehicle dynamics so that the UAV has to obey realistic physics along the path [29]. From there the algorithms seek to find the best flight path subject to different simulation parameters such as number of UAVs, number of targets, and whether or not there are hostile areas to be avoided. Examples of such studies can be seen in [30-32].

One main takeaway from these studies are that sufficient benefits can be realized over general path planning to warrant further study of implementing these optimal systems. The time to locate the target can be decreased due to the UAVs actively taking steps to minimize the uncertainty in target position instead of just flying a preset route. Geolocation accuracy can also be improved due to the UAVs actually flying to where uncertainty in the target position is minimized instead of just carrying out a preset path, which may be something as simple as a loiter that doesn't carry the UAV close enough to the target to get sufficient fidelity of direction finding measurements.

Another conclusion from these studies is that the literature is now fairly well developed for computing optimal path planning for a variety of generic scenarios. Examples of techniques that have been studied are nonlinear programing techniques and dynamic swarming techniques. These studies have been performed for a variety of basic situations, such as different numbers of UAVs, different numbers of targets, and including regions of denied airspace. Overall the current state of the literature in this field is developed but still with many opportunities to grow.

2.5.4 Further Research Opportunities.

After reviewing the state of the research that has been performed in simulated UAV geolocation, it can be seen that the work is fairly extensive and covers many different topics and trade-spaces. However, some areas remain where additional work can be done. One area of possible research includes looking at high uncertainty in the direction finding systems. The bulk of the prior research deals with systems that have a low amount of uncertainty in the bearing measurements, usually only one or two degrees. This produces results that are highly accurate and can be obtained with only a few measurements. The effect of highly uncertain systems with errors over ten degrees is much less well studied.

Another area of possible research is to include vehicle orientation and altitude in the studies. Most studies model the UAV as a point mass and simplify the geolocation problem to a two-dimensional problem. The possible effects of a yawing or rolling aircraft located at altitude are largely unknown.

An additional area of research would be to test the effectiveness of route planning algorithms in real-world environments. While simulations have shown the benefits and feasibility of optimal route planning algorithms, limited testing has been done on the physical implementation of these algorithms. Due to the large computational resources needed for such calculations along with the time delays associated with solving trajectory points, especially for a centralized system that is performing the calculations on the ground, much work remains to be done on addressing some of these issues.

Now that the theoretical aspect of UAV geolocation has been discussed, the current state of physically realized systems with flight testing will be examined. It will be seen that this aspect of the literature is much less well developed than the theoretical side.

2.6 Physical Testing of UAV Geolocation

Physical testing of UAV-based geolocation in a real-world environment has been severely limited compared to the large amount of work that has been done on the theoretical side. This is likely due to the fact that simulations generally come before physical testing as a way of proving the concept and refining the design to be tested. Also, the cost and effort required to perform real-world testing is much higher than performing computer simulations. This is due to having to acquire all of the physical hardware and components, assemble and transport the entire system, and having to find a location to fly the UAVs since many of the larger platforms are restricted in where they can be operated. This leads to many fewer studies being performed on real-world UAV geolocation.

Although testing is limited, one study recently published in 2013 by Bamberger, Moore, Goonasekeram, and Scheidt found that geolocation of a radio signal though the use of small, inexpensive UAVs was possible [21]. The goal of this study was to geolocate a ground-based radio signal with a team of cooperative UAVs. For the experiment, a mixture of commercial off the shelf (COTs) products and custom hardware was used. The actual UAV platform used was a Procerus Unicorn with a wingspan of only 5 feet. The UAVs in this test were cooperative and independent of ground station processing. What this means is that they were able to share their own measurements with

the other UAVs in order to speed geolocation time, and also all geolocation and path determining calculations were performed onboard the UAV without farming computing resources out to the ground station. The testing was broken into two parts. The first was to geolocate a hidden RF emitter with two UAVs flying a preset loiter pattern. The second part was to test the autonomy functionality and demonstrate convergence to a given emitter location.

In their study, the actual geolocation was performed using a pseudo-Doppler antenna array that produced LOB measurements in discrete 22.5 degree increments. The control was decentralized, meaning that each of the two UAVs possessed its own capability to process data and calculate where to fly to next. The UAVs were also cooperative and shared LOB measurements over a wireless local area network (WLAN). The loiter pattern flown was 550 meters from the emitter location, with an orbit radius of 50 meters. Using this set-up, an error of 60 meters between the estimated position and true target position was achieved. The error in position location over time for the study is shown in Figure 2.10. Simulation results compared to actual flight test results for the autonomy functionality test are presented in Figure 2.11.

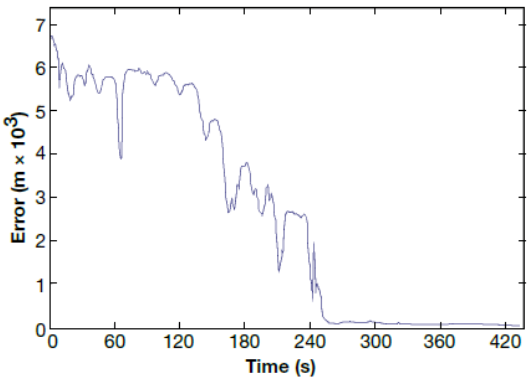


Figure 2.10: Reduction in error over time for flight test study [21]

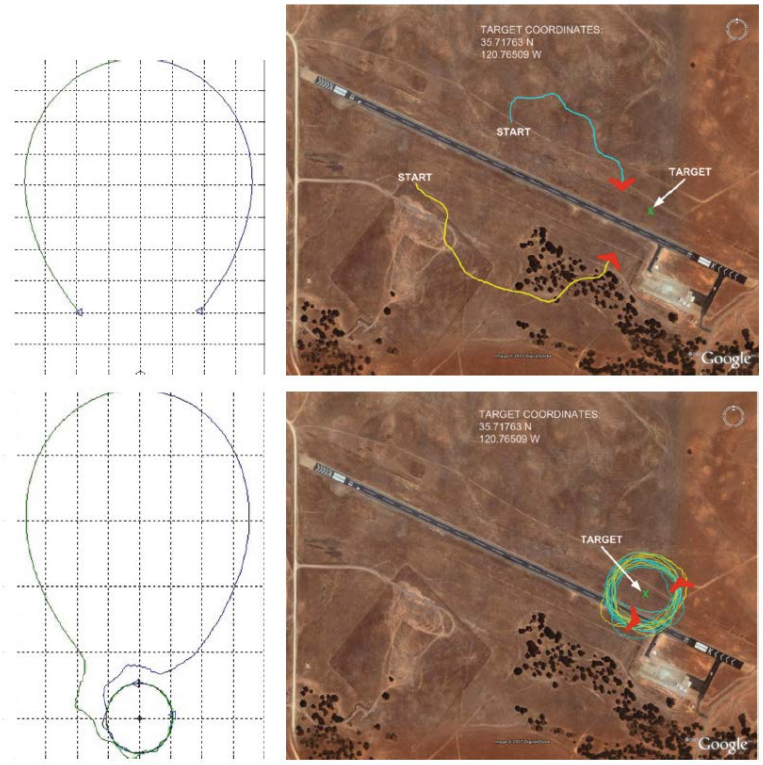


Figure 2.11: Simulation (left) compared to real-world testing (right) [21]

While this study does show the feasibility of a UAV-based geolocation package, there are a few drawbacks. One is that some of the hardware used is custom, and the processing algorithms are also proprietary. This could mean high cost and complexity of implementation which goes against the goal of having a low-cost and easy to implement system. Another drawback is that geolocation and path planning were not performed simultaneously, they were separated into two parts. It is unknown if doing them at the same time would require more computational resources than are available onboard the UAV, potentially leading to having to change the design from a decentralized one to a centralized one where calculations are performed on the ground station and then relayed to the UAV. The UAVs were also operated at low altitudes such that altitude effects were

ignored and the problem was assumed to be two-dimensional. The authors of the study suggest calibrating the LOBs with the vehicle orientation as a goal for future work to increase accuracy of the system.

2.7 Chapter Summary

In this chapter it was shown that the work done on the simulation side of UAV geolocation is fairly extensive. Many studies have been performed with a variety of methods and goals. These include different geolocation methods, different numbers of UAVs and targets, and different algorithms. However, the work performed on actual physical testing of UAV geolocation is very limited and provides much room for additional contributions.

In the remainder of this thesis, the methodology and results of the original research are detailed. This research focuses on expanding the physical testing of low-cost UAV geolocation and performing computer simulation studies to attempt to model the results of flight test and provide recommendations on how to improve the process of UAV-based geolocation.

III. Methodology

3.1 Chapter Overview

The purpose of this chapter is to provide an overview of the methods used to conduct both the geolocation simulations and the geolocation flight tests. The overall experimental set-up will be built up from the initial goals and design decisions, to the test procedures and data collected. The goal is to provide the reader with an insight into the work and decision making process that ultimately led to the results that will be discussed in Chapter IV. The flight test methodology will be presented first, and then the simulation methodology, as that was the chronological order that the work was performed.

3.2 UAV Flight Test Methodology

This section outlines the goals of the UAV flight test program, the hardware and software used to perform the tests, the overall procedure of the tests, and specifies what data was acquired throughout the tests. The goal is to provide an overview of the methods used to collect the data discussed in Chapter IV, and provide the motivation and thought process that led to the overall design of the system.

3.2.1 Goals of Flight Test.

The initial motivation for building and testing a UAV-based geolocation platform came from the small UAV design sequence offered at AFIT. The project for the three course sequence was to design and test a low cost UAV platform capable of locating a hidden RF emitter. There were other auxiliary goals of flight test including endurance and payload requirements, but this thesis will mainly focus on the geolocation aspect of flight

test, except where other goals may have motivated particular design choices. This goal of geolocating an RF signal from a UAV platform was likely chosen due to it being considered a challenging but feasible goal for the class, as well as by the interest of the professors in the emerging field of UAV-based geolocation. The overall expected outcome of the testing was an evaluation of the feasibility and accuracy that could be achieved with a low cost system, as well as to provide a basis of work that could be built upon and improved in the future. It should be noted that author worked as part of a team to design, build, and flight test the experimental UAV geolocation system. Post-processing analysis and simulations were performed independently by the author.

3.2.2 Flight Test Constraints and Assumptions.

Before the particulars of the UAV flight test methodology are detailed, it is helpful to discuss the constraints that were present from the start of the design process. The first constraint was that the design was limited to the use of COTs hardware. There was simply not enough time to design and develop any new UAV platform or direction finding device. The second constraint was on cost. The overall cost of the system was limited to $1500 dollars, excluding the UAV platform itself, which was provided. This was in keeping with the design goal of producing a low cost and easy to implement system. The final constraint was that flight testing could only be conducted at an allowable location, which was determined to be Camp Atterbury, IN. This was due to various safety and legal restrictions that prohibited the operation of the system near the local area. The effect of this constraint was that only ground testing could be performed

prior to the full-up flight test date, limiting the time available for troubleshooting, system improvement, and simulation.

Some assumptions also went into the design of the UAV geolocation system. These assumptions were presented as givens at the beginning of the design process. The first was that the hidden emitter would be located within a certain specified region of interest, and that it would not be physically covered or interfered with in any way. It was also assumed that the region of interest was located away from any major source of noise, and that the region would be clear of any obstructions that would prevent LOS between the emitter and receiver. Lastly, it was assumed that the emitter was broadcasting continuously and at a known frequency.

3.2.3 Flight Test Hardware.

The hardware used for flight test can be broken down into six main categories: airframe, propulsion, navigation & control, communications, RF payload, and ground station. This section will highlight some of the key components and decision making processes, but a full physical decomposition can be found in Appendix A.

First, for the airframe category, the small UAV class was provided with the option of selecting two different airframes. The first was a Super Sky Surfer, which was a small and lightweight UAV that was used by the previous years' class to carry out a different mission. The second airframe was a Sig Rascal 110, a larger and heavier platform with a roughly 9 ft. wingspan. This model retails at about $550 [33]. The decision was quickly made to go with this platform due to the larger payload capacity over the smaller model. The airframe used is pictured in Figure 3.1.

Figure 3.1: Sig Rascal 110 used for flight test

Second, for the propulsion system, the class was given two options. The first was a battery powered platform, while the second was a gas powered system. It was initially decided to use the battery powered variant, as this version was cleaner and simpler to operate than the gas powered version. However, as the weight of the components added up over time, and pressed to meet an endurance goal of 45 minutes, it was ultimately decided to switch to the gas powered variant for the increased payload capacity and endurance.

Next, the navigation and control components had to be acquired. The goal with the selection of these components was to use a cost-effective product with proven reliability. This decision was also coupled to what would be easily compatible with the ground station operation. This led to the selection of the ArduPilot Mega 2.6 autopilot with GPS [34], as it was widely used, within budget, and operated seamlessly with the already available Mission Planner ground control software.

The fourth main section of hardware selection had to do with communication devices. The full list of antennas and receivers used can be found in Appendix A. The main design goals with these selections were to pick devices that would allow for communication between the ground station and both the UAV autopilot as well as the RF direction finding system at a sufficient range, which was set as a goal of two miles.

The fifth main section of flight test hardware is the selection of the RF direction finding device. The class first examined a bow-tie antenna device that was tuned to the known frequency of the emitter. The specific model tested was the Handi-Finder radio direction finder [35]. This was deemed to not produce the desired level of accuracy out of the direction finding process. The search was then expanded for other low-cost, commercially available RF direction finding devices. Two models were decided on that worked off of the pseudo-Doppler principle that had produced reliable results for ground-based fox hunting applications[1], as well as being employed in the real-world flight test study discussed in section 2.6 of this paper. The first model was a Ramsey Doppler DF-1 kit which retails for about $150 [36], and the second was a KN2C RF Doppler direction finder, pictured in Figure 3.2, which retails for about $400 [37]. Although the KN2C system was more expensive, it possessed a greater angular resolution than the Ramsey system, specifically 10 degrees to 22.5 degrees. After performing ground tests on the two systems, it was determined that the KN2C system did indeed provided better results. It was decided that the KN2C system would be placed on the chosen airframe while the Ramsey system would be kept as a back-up.

[1] Fox hunting is a common term for a contest of how quickly transmitters can be geolocated by amateur ham-radio operators.

Figure 3.2: KN2C Doppler Direction Finding System [37]

The final piece of the hardware design was the ground station. The Mission Planner software was chosen for operating the UAV as it had a proven history of results, was readily available, and was free [38]. The software was operated on a stand-alone Lenovo laptop that was outfitted with transmitters capable of interfacing with the UAV. This set-up was ultimately operated out of a large trailer that provided protection from the elements, a mobile power generator, and a means of transporting the UAV to the testing facility in Indiana. A picture of the mission planning screen on the ground station can be seen in Figure 3.3.

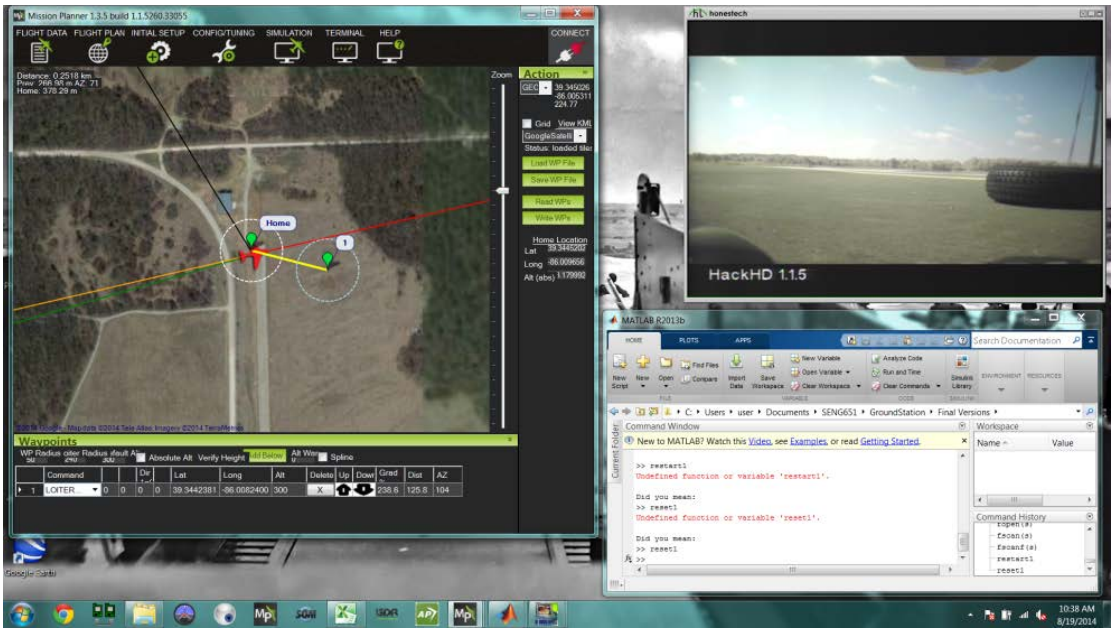


Figure 3.3: Mission planning screen on the mobile ground station

After all of the components had been acquired and tested, they had to be installed on the Sig Rascal platform. Everything fit within the vehicle, although space was tight. An area of improvement would be to determine a better layout of the components that allow for easier access between flights. For space savings, the actual direction finding hardware was removed from its black exterior shell shown in Figure 3.2 before being mounted inside the aircraft. One design consideration during this phase was to keep the center of gravity as close to the front of the plane as possible for stability and control, which was done by placing batteries in the very front and keeping everything else as far forward as possible. Another design consideration was where to locate the antenna array of the direction finding kit on the aircraft. Under the body or one of the wings was considered due to the antennas having a more direct line of sight to the emitter, but the array was eventually placed on top of the wing due to a variety of reasons including

structural issues, adequate space for mounting, and the fact that the antennas would drag on the runway if mounted directly underneath the aircraft. The antennas can be seen on top of the plane in Figures 3.1, and a shot of the internal components can be seen in Figure 3.4.

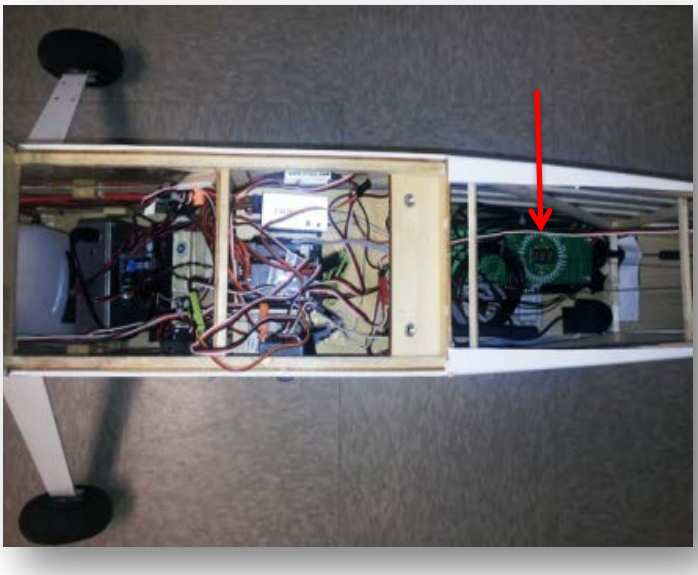


Figure 3.4: Internal components with the direction finder shown in green

3.2.4 Flight Test Software.

The flight test software programs used were the previously mentioned Mission Planner software, Google Earth, and codes written in Python and Matlab. The purpose of these codes was to take the output of the direction finder and plot the LOBs on Google Earth. A flowchart of the process is shown in Figure 3.6. The basis for these codes was designed by one of the course instructors, Dr. Colombi, and refined by team members 2nd Lieutenant Kyle Hathaway and Major Jason Png. A second code was used during post-processing to perform a least-squares geolocation estimate based on the collected LOBs, and then draw the position error ellipse associated with that data. The least-squares line

intersection code is shown in Appendix D while the error ellipse calculation code is shown in Appendix E.

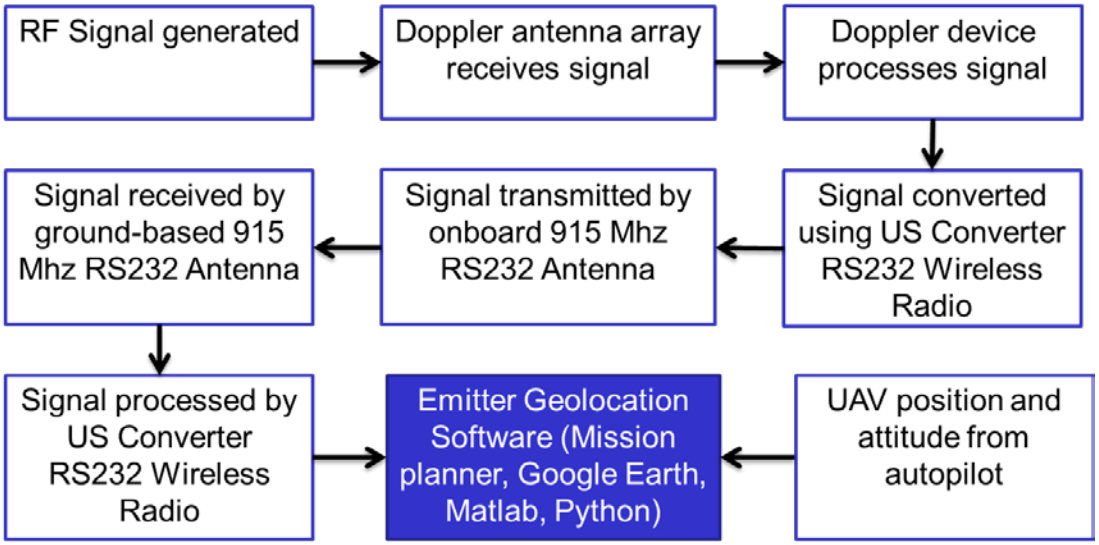


Figure 3.5: Flowchart of the general geolocation process

3.2.5 Ground Test Procedure.

Prior to taking the system to Camp Atterbury for the full-up flight testing, extensive ground testing was performed to examine the performance of the individual components of the system to attempt to get an idea of how well they would perform during the flight test procedure. All system components were tested, but the RF direction finding system will be focused on here.

The first focus of the ground testing of the direction finding kit was to take it out in an open field, dial it into the emitter frequency, and then leave it stationary as the emitter was carried around it and observe its performance. The goal of this was to determine the performance of the system under near ideal conditions, with the two systems located on the same plane (elevation) and in close proximity to each other. No

data was gathered from this besides observing how well the red light indicator on the direction finder matched the true position of the emitter. The red light indicator was the visual representation that showed the estimated position of the emitter relative to the direction finding kit. Since there were 36 equally spaced lights in a circle, each LOB measurement was then accurate to ± 10 degrees, assuming a perfectly functioning system with no noise or other sources of error.

The next focus of the ground testing was to increase the scale of the test and observe the performance of the system. This was coupled with testing the plotting script of the LOBs on Google Earth. This test was performed by affixing the direction finder to a car, and then driving around the local track with the emitter located in the middle. The results of this testing are discussed in Chapter IV.

The last major ground testing of the RF direction finding system was performed once the system had been mounted on the Sig Rascal. The purpose of this test was to verify everything was connected correctly and verify that the system produced similar results to those that were observed during the testing with just the RF package itself. No data was collected during this phase as it was observed that the system had similar performance both by itself and mounted on the UAV.

3.2.6 Flight Test Procedure.

The goal of the full-scale flight testing was to observe and test the performance of the system in the air and attempt to meet the outcome goals that were given at the start of the sequence. For the RF system this meant geolocating the target to within 40 meters using only radio direction finding. To test this capability, multiple flight tests were flown

for a total of five distinct geolocation attempts. These were conducted at different altitudes and different distances from the target. This was done to observe the effects that these parameters might have on the geolocation solution. No path planning work was done, as loitering patterns were the most simple and the main goal was to test the feasibility of the system, not yet to operate as efficiently as possible. Data was taken by manually running the script to take a LOB measurement and plot it on Google Earth. The results of these flight tests are discussed in Chapter IV.

3.2.7 Flight Test Wrap-Up.

Upon completion of the flight test phase, the main goal was to have acquired data on the performance of the RF direction finding system attached to the UAV platform. This meant getting the lines of bearing generated by the system, knowing the location of the UAV at each of those measurements, and knowing the true location of the emitter. By acquiring this data, the performance of the system could be evaluated and conclusions could be reached. The data that was collected along with a discussion of the results can be found in Chapter IV.

3.2.8 Flight Test Post-Processing.

The last area of work related to the UAV direction finding study was post-processing the results obtained during flight test. The first step in this analysis was to simply plot the data that had been collected during the testing. This involved taking the bearing measurements collected during test and converting the raw latitude and longitude data into relative distance data for easier analysis. This is done by taking the difference between a given point and a reference point, in this case chosen to be the known emitter

location, and then converting the distance from latitude and longitude to meters. This calculation is done in lines 34 - 35 of the Flight Test Analysis Code in Appendix C and shown below in Equations 3.1 and 3.2.

$$Relative\ X\ meters = (longitude - reference) * 85675.27 \quad 3.1$$

$$Relative\ Y\ meters = (latitude - reference) * 111030.25 \quad 3.2$$

After this was done, the next step was to plot the bearing measurements and from them determine the most likely target location. This was done using a least-squares approach that seeks to find the point that minimizes the distance from each bearing line.

Once the estimated target location had been determined, a measure for the quality of that estimation had to be determined. The standard method of using an error ellipse to portray this idea was chosen. The error ellipse shows the region that the target is likely to be contained within for a specified confidence interval. This means that for a lower confidence interval the error ellipse will be smaller, due to the decreased likelihood of the target being in that region. Conversely, a large confidence interval will yield a larger error ellipse. The generation of the error ellipse comes from looking at how well the data matches the estimated point. If all the bearing lines are close to pointing at the same location, the error ellipse will be small. If they point in diverging directions the ellipse will be larger. This gives a way to quantify the quality of the results, specifically by the size of the long radius of the ellipse. The calculation of the position error ellipse is shown in the Error Ellipse Calculation Code of Appendix E.

The second step of the flight test post-processing was to look at applying correction factors to the data in the attempt to improve the results. It was speculated that vehicle orientation played a role in the output of bearing measurements, and that correcting for those factors could potentially improve the results. To apply these correction factors, the vehicle orientation data from the onboard autopilot was correlated to the points at which bearing measurements were taken, and then the bearing measurements were shifted in various manners with respect to parameters like vehicle yaw angle and heading angle. This calculation of the shifted LOB measurements is shown in lines 56-57 of the Flight Test Analysis Code in Appendix C and presented in Equations 3.3 and 3.4 where θ is the bearing angle given by the direction finding device and φ is the correction factor angle to be applied, both in degrees. Adding in pre-programmed offsets of 45 and 90 degrees was also done to investigate if there was an inherent bias in the system which may have resulted in a systematic shift of all of the data points in the set.

$$LOB_x = magnitude * \cos\left(\theta_{RF\ Bearing} \pm \varphi_{Correction}\right) + relative\ X\ meters \qquad 3.3$$

$$LOB_y = magnitude * \sin\left(\theta_{RF\ Bearing} \pm \varphi_{Correction}\right) + relative\ Y\ meters \qquad 3.4$$

The last post-processing step was to sort the bearing measurements by the amount of aircraft roll present at the time the measurement was taken. The reasoning for this was that the direction finding unit was tested on the ground with the antenna array being perpendicular to the emitter, meaning that the array was not tilted at all with respect to the emitter. So in order to most closely replicate this condition, measurements taken

when a small degree of roll was present were identified and analyzed separately. For the purpose of this analysis, a range of ±10 degrees of roll was examined.

The results of these correction factors are presented in Chapter IV. The next section will now detail the methodology used to perform the simulated UAV geolocation studies.

3.3 UAV Simulation methodology

The purpose of this section is to outline the methodology used to create and analyze the simulated UAV geolocation studies. This will include examining the overall goals of the simulations, how the simulations were constructed, why they were done that way, and outline the trade studies performed to analyze the performance of the system.

3.3.1 Simulation Goals.

The first step in performing the analysis was to clearly identify the goals and expected outcomes of the simulations. The first goal of the study was to provide a realistic sense of the performance that could theoretically be achieved with a set-up similar to that used in the real-world flight testing that was discussed in the first part of this chapter. This meant using a platform that was relatively slow moving and whose geolocation accuracy is at a minimum on the order of 10 degrees, as this most closely resembled the set-up previously described in Section 3.2. This differs from most prior studies which assume a much lower uncertainty, on the order of one or two degrees.

An additional goal for the simulations was to provide some idea of the performance that could be expected under various parameters such as flight path, uncertainty in the LOB measurements, and the amount of tolerance in accepting or

rejecting outlying points. The goal was to analyze and look at the results of these different cases in order to provide recommendations on what combination of parameters would be best to try and employ in the physical system. The next section will discuss how these simulations were constructed.

3.3.2 Simulation Set-up.

The first step in constructing the simulations was to decide on how to model the actual geolocation process. The first idea was to use single bearing lines from the UAV to the target with some degree of uncertainty built in, and then use a least-squares method, which acts to find the point which minimizes the overall distance from each line to that point. That point would then be the estimated target position. This approach was decided against for two reasons. The first was that single bearing location fixing using a least-squares method is already well documented in literature, so any contributions would be minimal. The second reason was that this method provided poor initial results until the UAV had moved a substantial distance from its initial spot. This is because when the lines of bearing were generated close to one another, and the error was such that they diverged, the least-squares intersection between them was very near to the origin of those bearing lines. The UAV therefore had to build up sufficient distance between points to provide the necessary angular resolution that the least-squares intersection point was near the true target, and with slow moving UAV far from the target, that would be a non-ideal situation.

Instead of a single bearing geolocation method, a multi-line uncertainty cone method was used. In this method, a bearing measurement is generated and then two lines

representing the uncertainty in the measurement are formed on either side of that bearing measurement. So at each point instead of a single bearing line being generated, two lines forming a cone from the UAV outward to the target are generated. The geolocation is then performed by finding the points where these cones overlap, and then calculating the mean position of these points in space. This is similar to the approach shown in Figure 2.9, but instead of the overlapping area being used, it is only the corner intersection points. This provides more of a worst case scenario geolocation solution, but after studying the flight test results acquired previously in the study, it was determined that a worst case approach could be warranted.

The benefits to using this approach were twofold. The first was that this method seems to be less well documented in literature, leaving more room for possible contributions. The second benefit was that this cone method more accurately simulates what could be expected out of the real-world flight testing. The direction finding unit used to perform the geolocation in the flight test portion of this study only had an angular resolution of 10 degrees, which was displayed in the form of 36 LEDs in a circle, each LED therefore showing 10 degree increments. So even if this system was 100 percent accurate, the true line of bearing at each point would still only be known to within 10 degrees. So by modeling the bearing measurement as an uncertainty region, this would provide a better model of the system that was used in flight test. To model the actual uncertainty, a degree of randomness had to be coded into the simulation. This would simulate how real-world noise and sources of error would affect the results. For the random spread, a zero mean, normal distribution was chosen. This means that the samples were generally given in a bell curve shape around the true position, and that

larger outliers were less frequent. This randomness is generated in line 232 of the Geolocation Code in Appendix B. The calculation of the true bearing angle, β, from the UAV to the target with this randomness built in is given in Equation 3.5 where *x* and *y* are the coordinates of the UAV and the target. The angle of each of the cone lines in given in Equations 3.6 and 3.7, where σ is the angular resolution of the measurements, i.e. 10 degrees in the case of the Doppler direction finding unit used during test.

$$\beta = atan2(y_{true} - y, x_{true} - x) + error \tag{3.5}$$

$$\beta_{Left} = \beta + \sigma \tag{3.6}$$

$$\beta_{Right} = \beta - \sigma \tag{3.7}$$

After the geolocation method had been determined, a method of characterizing and showing the performance of the system had to be constructed. The most widely used method of doing this is through the error ellipse. The idea of the error ellipse is that it shows the region that the true measurement is most likely to be in based on the observed input measurements. The size of the error ellipse will depend on the accuracy and consistency of the input measurements, as well as the specified confidence interval. For this research, a confidence of 50% was chosen. Choosing a higher or lower confidence would result in a larger or smaller ellipse respectively, but the shape of the ellipse would remain unchanged. Ultimately, the error ellipse is a physical representation of the covariance of the data. The covariance in this case quantifies the degree to which variables are correlated in two dimensional space. The covariance looks at the variance, which is a measure of how that each variable is distributed, both with itself as well as with the other variables in that space. The error ellipse then gives a physical

representation of this idea in two-dimensional space [39]. For example, an elongated ellipse would show a high degree of variance in one axis of the measurements but not the other.

Once the method of geolocation and method of determining the results had been decided, the specific details of the simulations had to be sorted out. The first was how to quantify the results for comparison among tests. The method chosen was the length of the semi-major axis of the error ellipse. This single number provided a good sense of the accuracy of the geolocation and could easily be compared between cases.

The next specific detail was how to simulate the UAV motion in the study. Since all analysis was two-dimensional, the method of steering the UAV over a path was to use the heading angle of the plane. This gave a simple method for controlling the UAV. A velocity of 35 miles per hour was used in the study as a representative value for a small, slow moving UAV. This would allow for realistic times and distances to be simulated.

The final main detail of the study was the idea of filtering, or sorting, the data to get rid of outlying points. Due to the uncertainty and error in the bearing measurements, some measurements are bound to be closer to the true location than others. By throwing out outlying points and only keeping the points more in agreement with other points, it was anticipated that this would produce more accurate results. To accomplish this, a filter had to be included in the simulation. The methodology behind this filter is as follows. The UAV would be allowed to gather a certain number of initial measurements, set to five in this study, in order to build up an initial database of results. Once the initial data had been collected, the standard deviation of the intersection points was calculated. From then on, the intersection points were compared to this standard deviation, and points too

far outside of it were discarded from the geolocation. This calculation is given in lines 318-333 of the Geolocation Code in Appendix B and shown in Equation 3.8 where *n* is the multiplying factor on the standard deviation. The goal was to keep only the points closest to the estimated target position in the aim of improving overall geolocation accuracy. The criterion for throwing out intersection points is shown in Equations 3.9 and 3.10, where *inter* is the coordinates of the intersection point. If either Equations 3.9 or 3.10 were true, the point was thrown out.

$$Tolerance = |n * standard\ deviation(intersection\ matrix)| \quad 3.8$$

$$if\ inter > mean\ intersection + tolerance \quad 3.9$$

$$if\ inter < mean\ intersection - tolerance \quad 3.10$$

Once the specific details of the simulation had been determined, the overall test matrix and variables to be examined had to be laid out. This is detailed in the next section.

3.3.3 Simulation Test Cases.

Once the basic simulation had been built, test cases were then run to evaluate different parameters of the simulation. The test cases were broken up into three main categories: eight different paths, three different angle uncertainties, and three different filters for a total of 72 different cases.

The first main area of study was the effect of path on the geolocation results. From literature it is known that flying closer to the target is generally the best way to increase geolocation accuracy. From this idea, eight basic flight paths that could easily be

performed by the real-world UAV without any optimal path planning calculations necessary were constructed. These paths were a straight line both near and far from the target, a loiter around the target with a small, medium, and large radius, a loiter away from the target with both a small and large radius, and a buttonhook pattern where the UAV flies toward the target but at an angle so as to continue gathering measurements from different angles. From the literature on optimal path planning, it was projected that this path would give the best results in the shortest amount of time.

The next area of iteration was on the degree of uncertainty in the geolocation measurement. For this, one small angle of uncertainty was chosen, 1 degree, to represent what is likely the best case scenario of the geolocation and to serve as an ideal baseline. Two larger angles of uncertainty were also chosen, 5 and 10 degrees, to simulate what is more likely achievable in the physical system. The amount of degraded performance could then easily be compared to the more ideal scenarios.

The last area of iteration was on the filter built into the simulation. This filtering of unreliable points is likely something that could be easily implemented in a real system, so the effects of different tolerances warranted study. For this research, three different filters were studied. The first was no filter at all, meaning all intersection points would be included in the geolocation. The second was a filter that threw out points that were outside of one standard deviation of the intersection points. The last filter threw out points that were outside one-half of one standard deviation of the points, meaning it was the most strict.

The last important idea of the simulations was the idea of reproducibility. Since all of the results could vary based on the randomness introduced into the system, it was

desired to know how these results could change based on the randomness of the system. In order to best analyze this effect, Monte Carlo simulations were performed for each test case. Monte Carlo simulations allow a variable to fluctuate each iteration of the test. So for this study the variable fluctuating was the bearing measurements, which came from the randomness introduced into that measurement. This would allow for the system performance to be evaluated over many runs as a way of determining the bounds of the expected performance. For this research, the number of Monte Carlo simulations performed for each test case was chosen to be 100.

3.3.4 Simulation Wrap-Up.

Upon exiting the simulation phase, the desired results were to have error vs. time plots for each test case. This would allow for comment on the effect on performance and recommendations to be made based on those. These results are discussed in Chapter IV. The Matlab code used to perform the simulations outlined in this chapter is given in Appendix B.

3.4 Chapter Summary

This chapter has detailed the design methodology behind both the simulated and real-world testing of UAV geolocation. The goal was to provide an overview of what was done to arrive at the results and the rationale for why things were done that way. The exit criteria to evaluate the performance of the systems is ultimately the accuracy of the geolocation process, quantified by the size of the position error ellipse. The next chapter presents and analyzes the results obtained from the methods discussed in this chapter.

IV. Analysis and Results

4.1 Chapter Overview

This chapter presents the results of the experiments discussed in Chapter III. The goal of this chapter is to provide all of the relevant results as they pertain to the goals of this research effort that were obtained from the real-world UAV flight testing as well as the computer simulations. These results are presented and then discussed in relation to the methods of performance introduced in Chapter III, as well as the limitations outlined in Chapter I. The purpose is to present what was accomplished and discuss the accuracy and implications of it in context of the scope of the work.

4.2 Flight Test Results

The first results to be presented are those from the real-world flight testing, as these were obtained first chronologically and helped shape some of the design decisions of the simulations. These results are broken down into three main sections: ground testing, flight testing, and post-processing.

4.2.1 Ground Testing.

As discussed in Chapter III, the ground testing of the Doppler direction finding system proved to be reliable in accurately showing the direction of the emitter. No data was gathered from the walk-around tests but it was observed that the direction finder generally pointed in the correct direction of the emitter. Although overall the performance was solid, the system output was prone to bouncing around, even when both the emitter and receiver were stationary. This was characterized by the LED occasionally

jumping to other points in a seemingly random fashion before returning to the correct location. The LED would also bounce back and forth between adjacent LEDs that more or less pointed to the correct emitter location. This was likely due to environmental noise and multi-path issues, as well as the inherent limitations on accuracy present in the system.

A scaled up version of ground testing was also done with the direction finder mounted to a car and then driven around the emitter. Results of this test were promising and gave an error in emitter location of only 18.8 meters. The results of this test are shown in Figures 4.1 and 4.2.

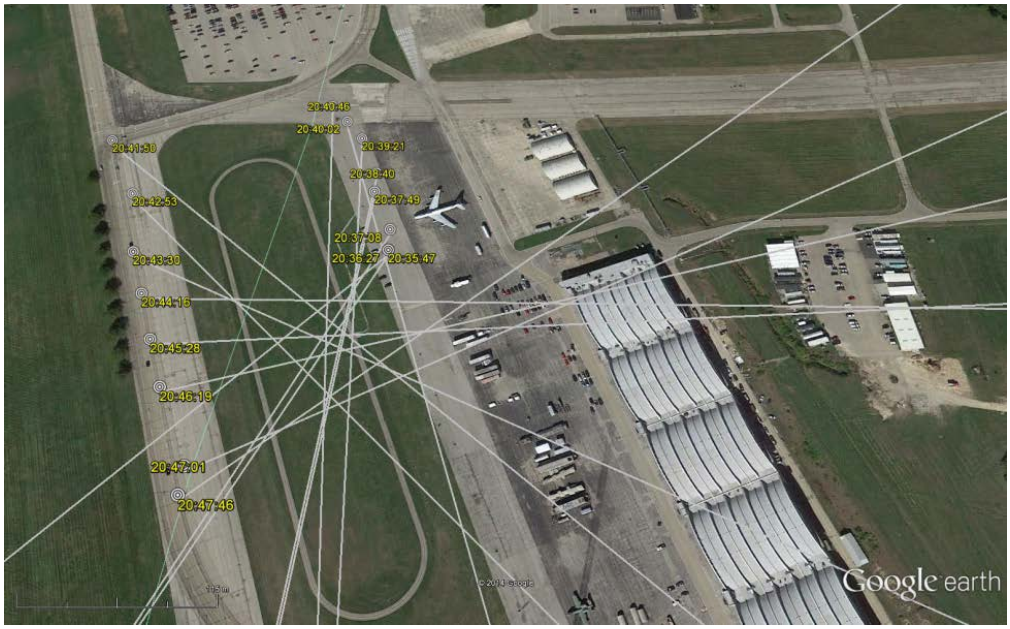


Figure 4.1: Google Earth overlay of direction finding ground test

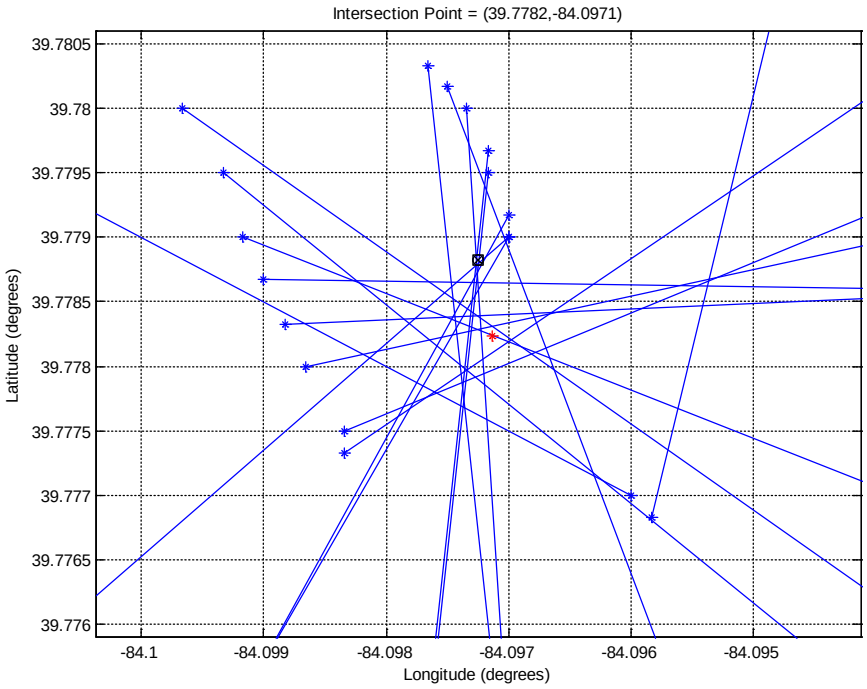


Figure 4.2: Estimated target shown in red, true target by a black circle

4.2.2 Flight Testing.

The next portion of flight test results come from the flight testing phase itself. It was during this phase that the geolocation system was flown on the UAV and put to the test in a real-world environment. The UAV was flown in the general shape of a circular loiter overhead, near, and far from the emitter, and at low and high altitude for the near and far locations, for a total of five test flights. The plot for one test flight, the overhead case as designated in Table 4.1, is shown in Figure 4.3. This depicts the location of the UAV at each measurement, the LOB measured at each point, the estimated target position in red, the actual target position in black, and error ellipses for both 50% and 95% confidence intervals. The overall results are presented in Table 4.1.

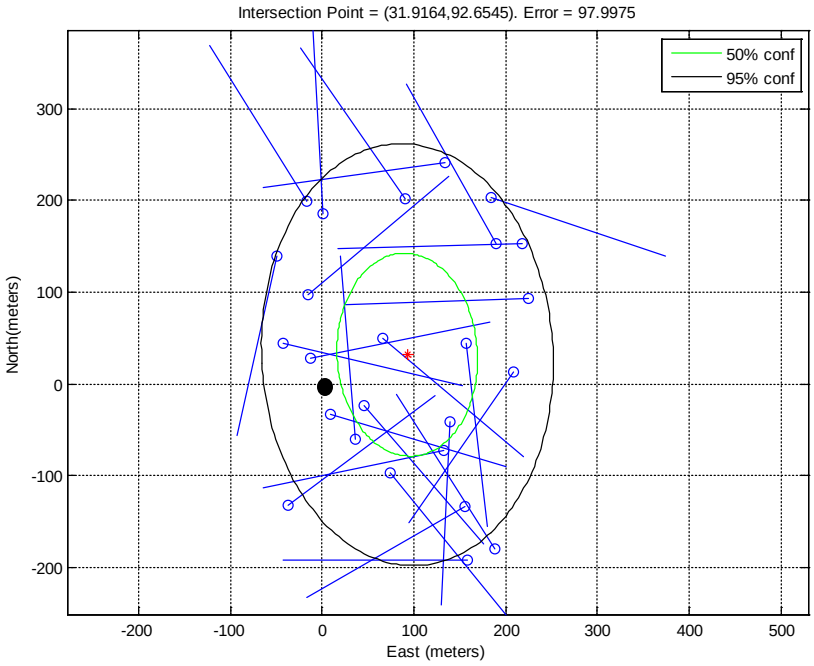


Figure 4.3: Geolocation results from overhead flight test

Table 4.1: Overall results of the flight test program

Horizontal Distance between RF Emitter and UAV (m)		**Average Altitude of UAV (m)**		**Error Between Estimated and True Position(m)**
Close	51	High	115	109
Far	205	Low	87	224
Far	232	High	111	247
Close	164	Low	54	180
Overhead	35	Overhead	78	98

4.2.3 Post-Processing.

The last piece of the flight test results came from post-processing the results. It was hypothesized that some sort of correction factor could be found that when applied to the raw data would provide better results. It was thought that this correction factor might come from correlating the UAV orientation with the bearing measurements. The most likely effect on the bearing measurements was thought to likely come from the yaw angle of the aircraft. The effect of subtracting off the yaw angle from the bearing angle is shown in Figure 4.4 for the same overhead flight test shown in Figure 4.3. The blue lines are the original data while the red and green lines represent the modified results.

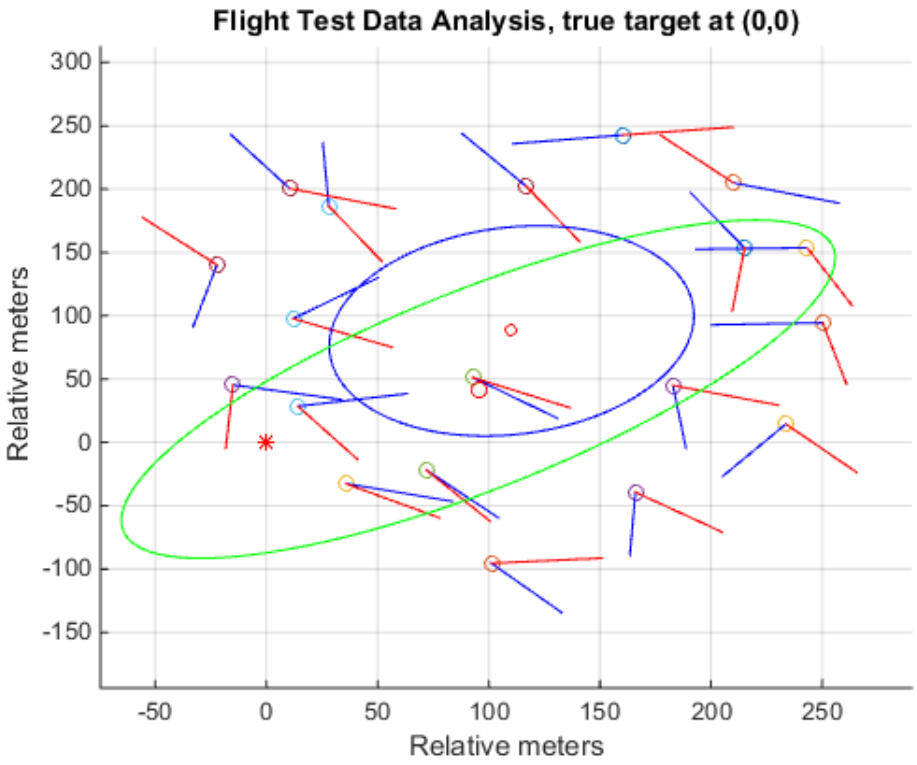


Figure 4.4: Yaw correction applied to original overhead flight test results

A multitude of other correction factors were examined. These included looking at the RF heading angle, and autopilot heading angle, along with combinations of all three. Adding offsets of 45 and 90 degrees was also looked at.

The last aspect of flight test post-processing was to look at the data acquired during times when the aircraft was experiencing a small amount of roll, specifically ±10 degrees. This resulted in a total of 15 points that fit the criteria out of a total 110 collected. Results for two of the flight tests are shown in Figures 4.5 and 4.6 below. Red lines indicate bearing measurements taken during times of 0 to 10 degrees of roll, red dashed lines are from -10 to 0 degrees of roll, and blue lines are from measurements outside of those ranges. The green error ellipse shows the uncertainty region for only the measurements in red.

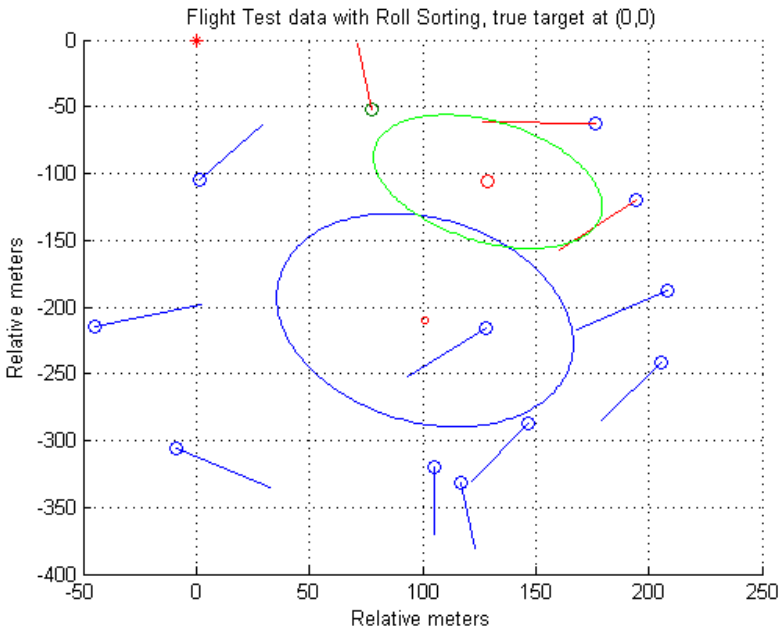


Figure 4.5: Roll sorted data correction from the first flight test

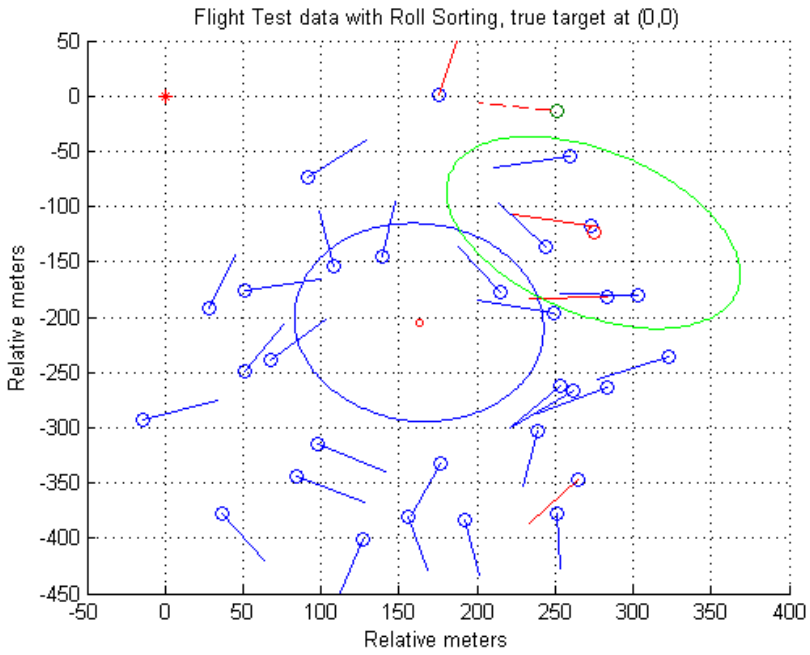


Figure 4.6: Roll sorted data correction from the third flight test

4.3 Flight Test Discussion

Now that the overall results of the flight test program have been presented, the accuracy and implications of the results will now be discussed. This is again broken down into the three main phases of the testing.

4.3.1 Ground Testing.

The results of the ground testing showed that the Doppler direction finding unit had the ability to accurately locate the RF target. Although the system was not perfect and suffered from some erratic performance, the overall results of the ground testing suggests that the pseudo-Doppler device is capable of delivering good results, as evidenced by the converging bearing lines shown in Figure 4.1. These results were as expected due to this device being commonly deployed by ground based fox hunting teams, with results that can be attested to.

4.3.2 Flight Testing.

Examining the results of the flight test program presented in Table 4.1, many conclusions can be drawn. The first is that the performance of the system was not good. For the overhead and close loiter at high altitude cases the true emitter position was within the 95% confidence region of the error ellipse, but for the other three cases it was not. And for how close the UAV flew to the emitter, all of the errors were much larger in comparison. There seem to be pockets of repeatable results, where measurements taken in the same general area all point in the same general direction. This suggests that the measurements are not completely random, that some repeatable process was occurring. But looking at the data overall it is very inconsistent, and generally predicts the target to be inside the loiter path of the UAV. This fact coupled with the large errors observed suggest the system was not functioning as intended, and certainly not giving results anywhere close to what was observed on the ground.

The implication of these results is that there is some aspect of taking the direction finder and putting it on a UAV at altitude that causes substantially degraded results. The only meaningful differences between the ground and flight tests seem to be having the direction finder located at altitude, and having it in a variety of orientations as the aircraft yaws and rolls around. One area of specific interest is in the effect of tilting the antennas relative to the emitter. Due to the high angles of roll encountered during flight test, shown in Figure 4.5, the antenna array experiences a great deal of tilt relative to the emitter. Studying the effect of this movement is an area for possible future work.

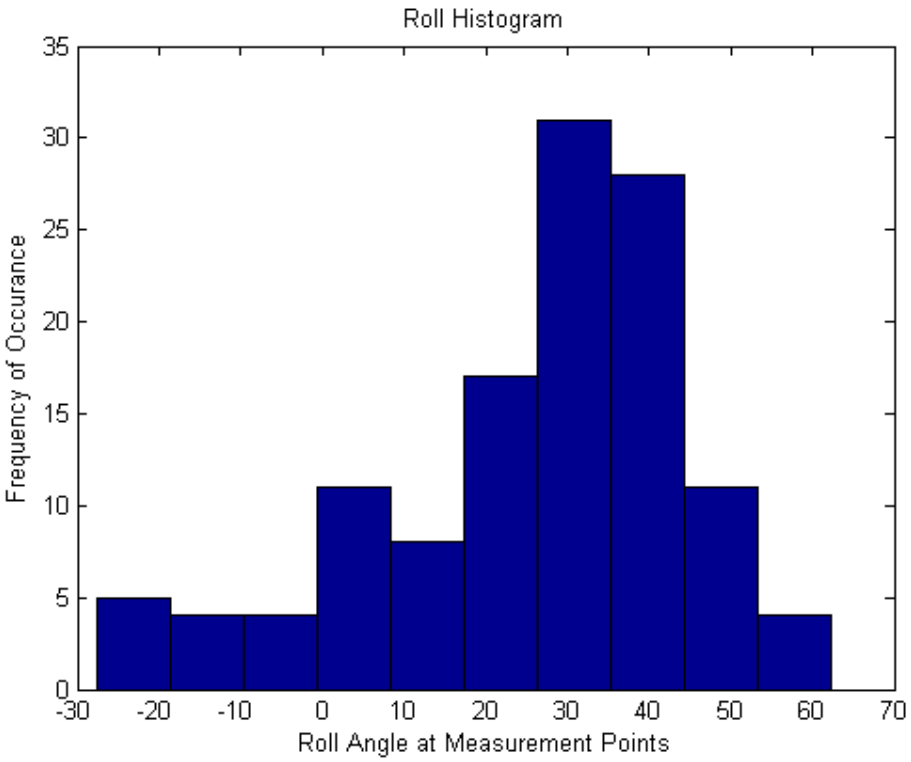


Figure 4.7: Roll angle in degrees at points when LOBs were recorded

Overall the results suggest that in its current form this system does not provide good enough results to accurately geolocate the target to any degree. This means that until the source that is causing the large discrepancy between ground and air testing is discovered, this system is unlikely to produce accurate results. However that does not mean the results are meaningless. It has been shown that the straightforward approach of mounting a direction finder on a UAV and flying that set-up is unlikely to produce the same kind of results observed on the ground. Within the limitation that flight testing was only conducted in one trip to Camp Atterbury, leaving no real time to study the effects and attempt to make improvements to address them, this testing was successful in that it

provided a baseline for expected results with the system as-is, and also provided topics of future study that can be done to improve the performance of the system.

4.3.3 Post-Processing.

After a series of post-processing corrections were made trying to find a meaningful relationship between bearing angle and other parameters like yaw angle, RF heading angle, and GPS heading angle, it was ultimately concluded that no such meaningful relationship could be found to exist. Variations of adding and subtracting these angles in different combinations both with and without preprogrammed offsets of 45 and 90 degrees produced no results that were clearly better than the original flight test results. The post-processed bearing measurements all still seemed to be strewn about randomly. Even when looking at the measurements sorted by small amounts of roll, no improvements were made. The sample size was small, with only 3 to 5 measurements for most tests and none for the overhead test, but in three of the four flight tests that had viable points, looking only at these points made the estimated target position worse. This suggests that more testing would be needed to characterize the effect of roll on the performance of the direction finding system.

One limitation was that the line intersection code used to obtain the best guess at the target location treats the bearing lines as extending infinitely in both directions. In reality this should not be the case, the direction finder gives directional bearing lines, meaning that 0 degrees and 180 degrees are completely opposite directions, not the same as this code assumes. Another limitation was that the altitude of the UAV was ignored, the problem was treated as a two-dimensional one.

The implications of the post-processing are that the collected data might just be wrong, and that no single correction factor will fix it. However, this analysis did not look into the effect of altitude on the geolocation. That avenue may still provide a way forward in analyzing and improving the data.

4.4 Simulation Results

This section will provide some of the highlights of the simulation results obtained by the methods outlined in Chapter III. First, some representative path plots are shown in Figures 4.8 and 4.9. These show the location of the UAV over time by a green line, the uncertainty cone lines from the UAV to the target in red, the estimated target position given by a red star, the bearing intersection points by blue dots, and the position error ellipse is shown in blue. The true target location remains at (0,0) for each trial and is marked in green. Figure 4.8 depicts the UAV flight path for the 50 meter radius overhead loiter while Figure 4.9 shows the flight path for the straight line pass. Both figures show the progression of the error ellipse and estimated target position as the UAV travels along its path.

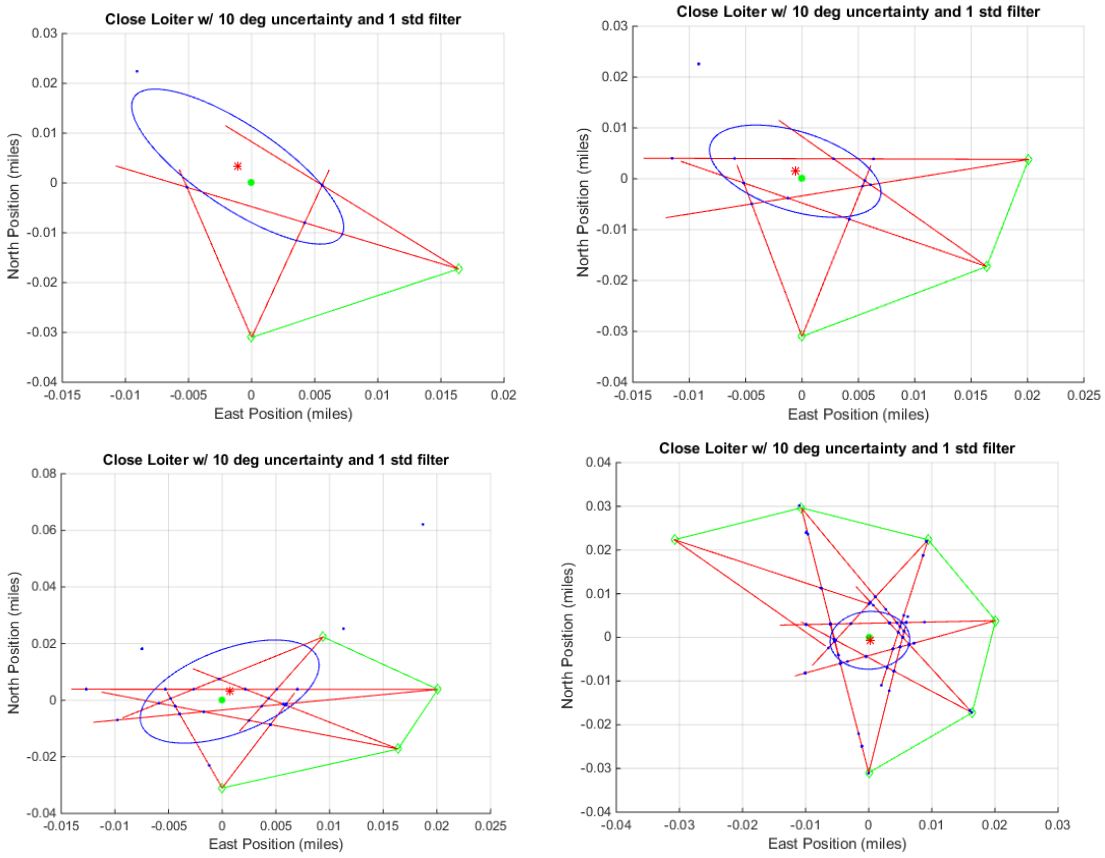


Figure 4.8: Path and error ellipse progression for 50m loiter case

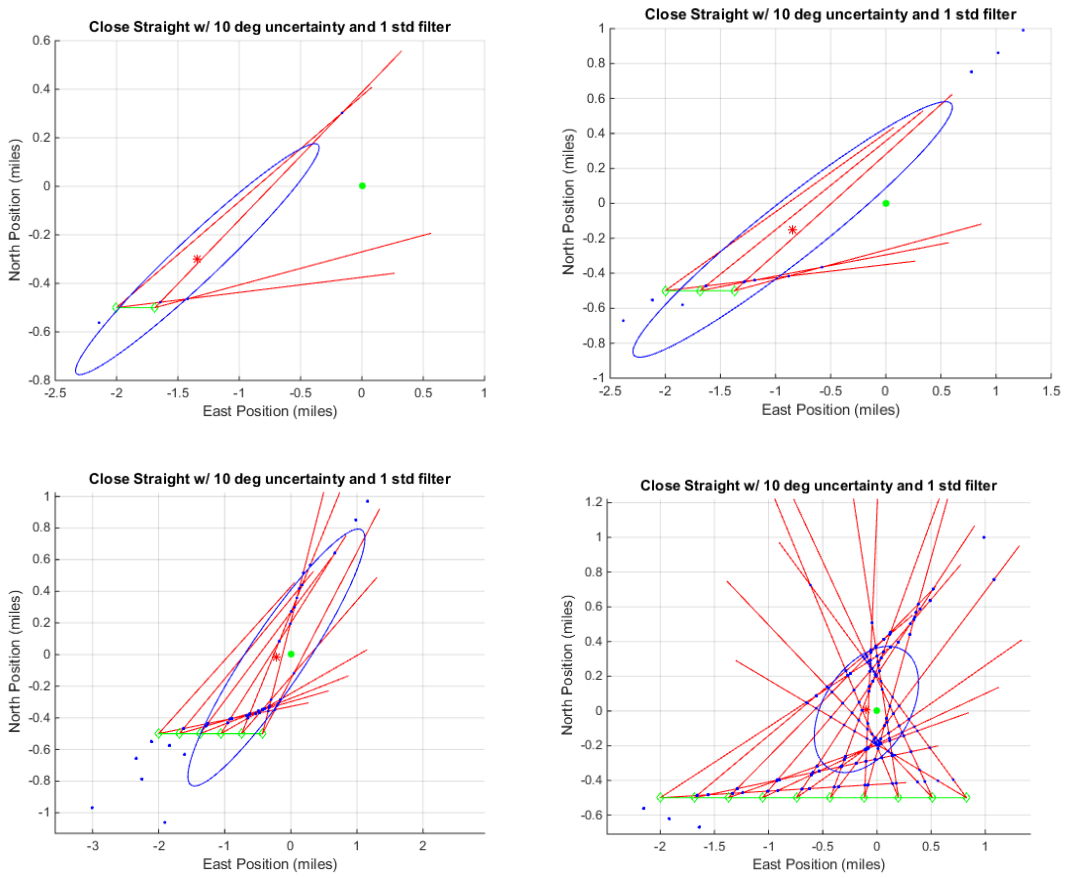


Figure 4.9: Path and error ellipse progression for straight line case

Now that typical path and error ellipse progression plots have been shown, some typical error over time plots will be presented that will aid in the discussion of the results in the next section. Figures 4.10 and 4.11 show the results of the Monte Carlo simulations for a couple test cases. Figures 4.12 and 4.13 show similar results for different test cases presented in an error bar format, where the maximum, minimum, and average (mean) values of the Monte Carlo simulations are shown.

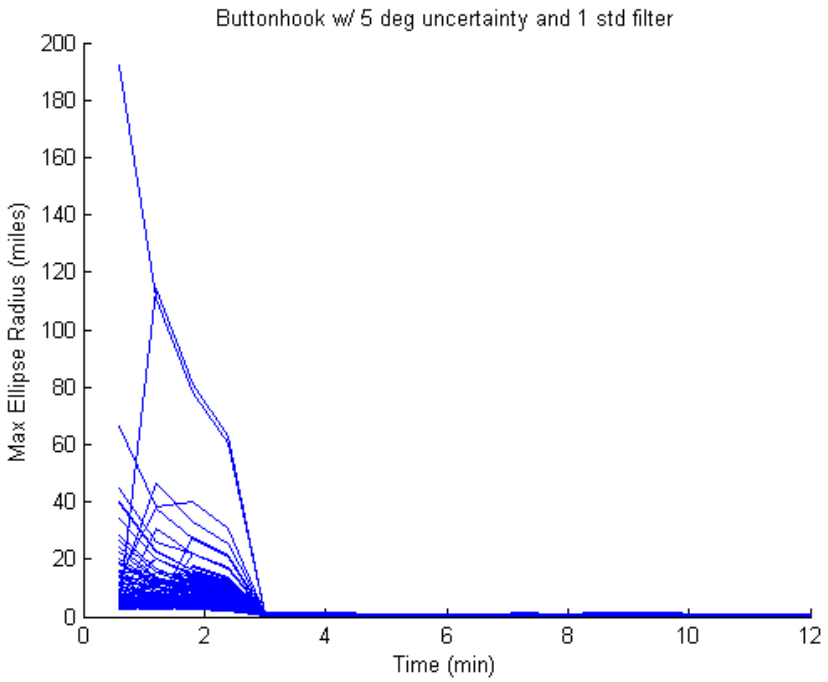


Figure 4.10: Error progression for buttonhook scenario

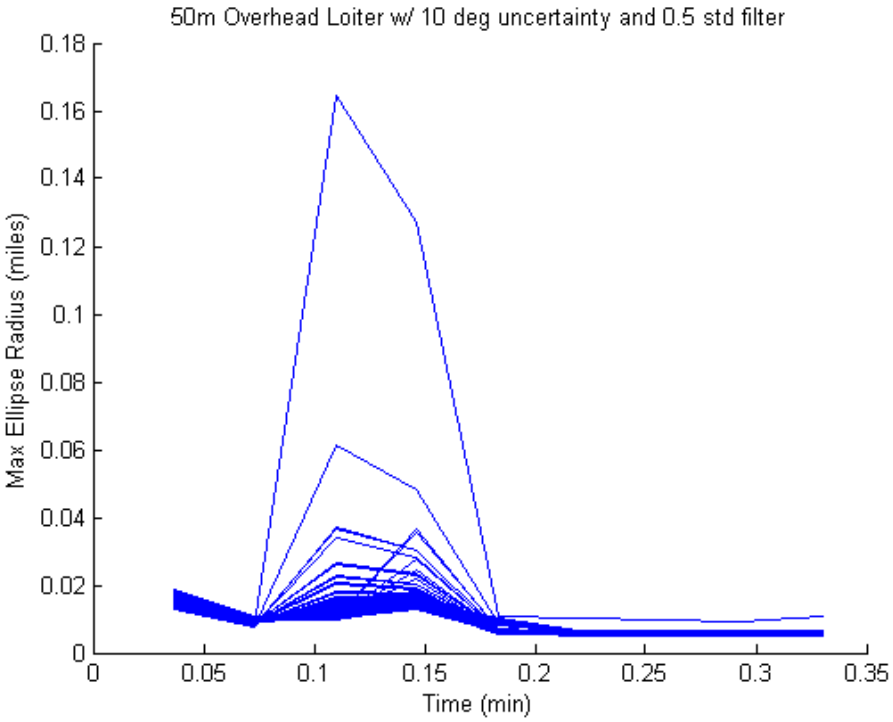


Figure 4.11: Error progression for 50m overhead loiter scenario

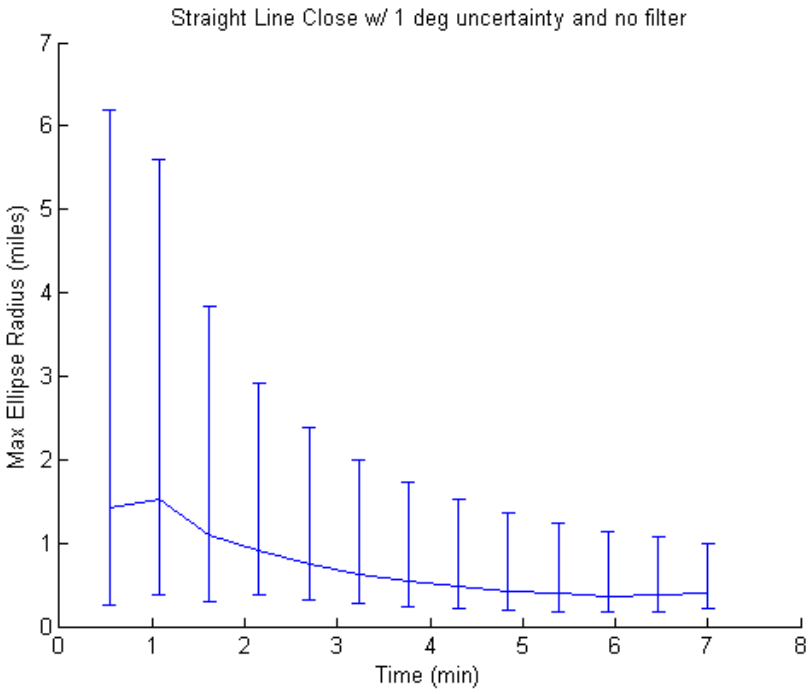


Figure 4.12: Error bars over time without a filter

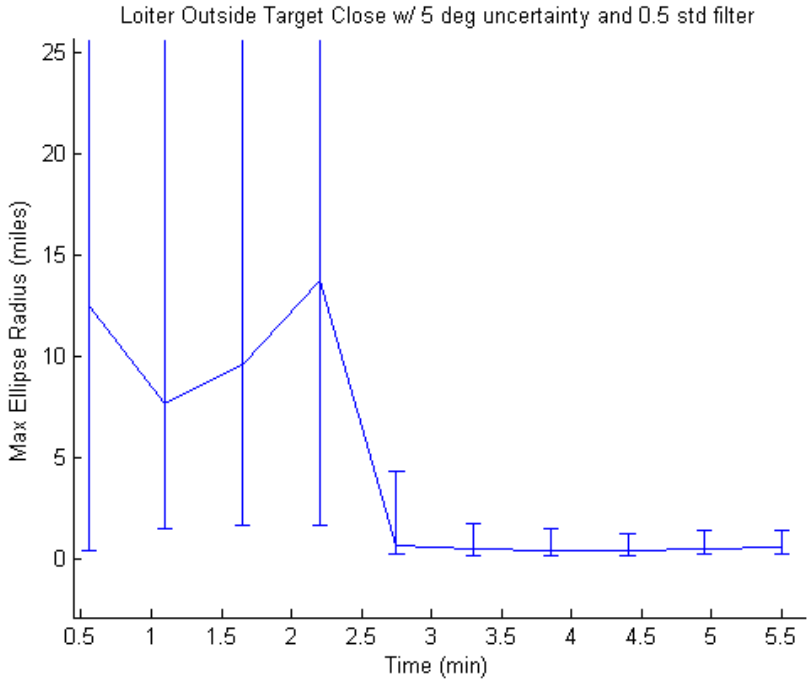


Figure 4.13: Error bars over time with a filter

Now that some representative visuals have been shown, the results from each test are compiled and presented in Table 4.2. This table shows the minimum error ellipse radius obtained for each trial, the distance in meters from the estimated target position to the true position at the point where the error ellipse radius was a minimum, and the time in minutes at which the error ellipse radius first drops under 0.1 miles, if it occurs for that test case. In many cases the error never dropped below this threshold, as indicated by a '-' in Table 4.2. It should also be noted that all results are taken from the average Monte Carlo value, as shown in Figures 4.12 and 4.13.

Table 4.2: Simulation Results for all 72 Test Cases

Flight Path	Uncertainty (Cone) Angle in Degrees	Intersection Filter (standard deviations)	Min Ellipse Radius (miles)	Distance from Estimated Position to True Position (meters)	Time at which Error Ellipse < 0.1 in minutes
Straight Line w/ 0.5 mile offset	1	none	0.441	36.9	-
		0.5	0.049	35.1	3.7
		1	0.066	11.7	4.3
	5	none	1.312	60.5	-
		0.5	0.165	60.1	-
		1	0.261	83.4	-
	10	none	1.464	1456	-
		0.5	0.296	365.5	-
		1	0.457	313.3	-
Straight Line w/ 2 mile offset	1	none	0.654	72.3	-
		0.5	0.092	164.5	5.4
		1	0.158	74.9	-
	5	none	2.858	3180	-
		0.5	0.546	320.5	-
		1	0.696	62	-
	10	none	2.411	2560	-
		0.5	0.887	574.4	-
		1	1.457	2874	-
Loiter Around Target w/ 0.5 mile radius	1	none	0.013	18.1	0.55
		0.5	0.014	4.4	0.55
		1	0.015	20.1	0.55
	5	none	0.057	12.1	1.1
		0.5	0.054	11.1	1.1
		1	0.054	11.1	1.1
	10	none	0.138	34.3	-
		0.5	0.136	20.3	-
		1	0.128	25.8	-
Loiter Around Target w/ 50 meter radius	1	none	0.0008	1.88	0.04
		0.5	0.0005	0.418	0.04
		1	0.0007	0.659	0.04
	5	none	0.0038	0.816	0.04
		0.5	0.0027	2.84	0.04
		1	0.003	2.57	0.04
	10	none	0.0084	3.02	0.04
		0.5	0.0054	2.06	0.04
		1	0.007	7.27	0.04
Loiter Around Target w/ 2 mile radius	1	none	0.167	59.0	-
		0.5	0.033	36.3	3.6
		1	0.057	26.9	4.1
	5	none	3.310	285.4	-
		0.5	0.286	75.1	-
		1	0.367	459.9	-
	10	none	1.710	1867	-
		0.5	0.501	509	-
		1	0.716	160.5	-
Loiter 1 mile outide target w/ 0.5 mile radius	1	none	0.282	49.8	-
		0.5	0.078	94.2	3.3
		1	0.148	134.1	-
	5	none	4.830	110	-
		0.5	0.392	59.9	-
		1	0.633	743	-
	10	none	1.604	1919	-
		0.5	0.892	213.1	-
		1	1.454	598.5	-
Loiter 4 miles outside target w/ 2 mile radius	1	none	9.110	480.4	-
		0.5	0.651	111.8	-
		1	0.945	459.4	-
	5	none	5.380	2231	-
		0.5	2.204	513.2	-
		1	3.238	1499	-
	10	none	5.134	1079	-
		0.5	2.803	5992	-
		1	2.889	5387	-
Buttonhook pattern beginning 2 miles away	1	none	0.126	38.15	-
		0.5	0.015	5.00	3.47
		1	0.024	10.83	3.13
	5	none	1.999	20.78	-
		0.5	0.173	34.97	-
		1	0.215	69.33	-
	10	none	1.318	1573	-
		0.5	0.303	78.36	-
		1	0.428	523.8	-

4.5 Simulation Discussion

From the results obtained from the simulation test cases, multiple conclusions can be drawn. The first is that flying closer to the target does indeed increase accuracy of the geolocation. Looking at Table 4.2 it can be seen that for paths that were similar except in distance from the target, the case were the UAV was closer to the target produced a smaller error ellipse radius in every case. This is an expected result since the further away the UAV is from the target, uncertainties in the LOBs propagate over a larger distance and create larger errors in the estimated target position.

The second conclusion from the simulation studies is that the presence of a filter improves the geolocation result in that the error ellipse radius is much lower. This makes sense since the filter removes points that are outliers from the current data set, making the grouping tighter and therefore producing a smaller error ellipse. However, an interesting observation is that the filter did not always improve the distance from the estimated target position to the real target position. In fact, this distance seemed to bounce around between runs and was at times worse when the filter was added. A likely explanation for this is how the filtering was implemented. A baseline of intersection points was built up over the first five measurements, and then future measurements were compared to that baseline. So if too many of these initial measurements were poor, the algorithm may start focusing down on the wrong point. The result is that sorting out bad data points in this method improves precision, but not always accuracy. However, in a real world scenario the true target position is unknown so there is no truth value to compare to in order to sort by accuracy. Overall, more work would need to be done to investigate different methods

of sorting points and to characterize the effect of filtering out points and how it affects the speed and accuracy of geolocation.

Another observation is that wild fluctuations can occur at the beginning of the geolocation process. With only a few measurements, the result is highly sensitive to any large outliers that can occur with lines that are nearly parallel. This causes some very large error ellipses that are elongated along the lengths of the lines as shown in Figure 4.9. This supports the notion that the UAV must fly at an angle to the target to acquire more information about the location of the target, and that remaining in place or flying directly at the target provides little to no new information. This is shown in Figure 4.8, as the UAV flies at different angles to the target, the ellipse size is decreased and becomes more circular shaped, instead of highly elongated in one axis.

Another interesting observation is that for the case that was most similar to the flight testing done, the 50 meter radius loiter with 10 degree uncertainty and no filter, very good results were still obtained, with the target being located to within 10 meters of its actual position. This suggests that if the experimental set-up could actually produce the desired accuracy of ±10 degrees, the results could still be very accurate.

The implication of these results is that they show geolocation with a larger uncertainty direction finder, on the order of ±10 degrees, is adequate to produce reliable and accurate geolocation results if the UAV can fly sufficiently close to the target, depending on the desired geolocation accuracy. However, the challenge is creating a low-cost system that is actually capable of providing that sort of accuracy.

4.6 Chapter Summary

This chapter has detailed the results of the work that was performed on simulated and real world UAV geolocation. The physical results showed that the geolocation system produced promising results during the ground tests, locating the emitter to within 18 meters. However, the flight testing showed that integrating the system onto an aerial platform and then performing the geolocation process from the air gave poor results. This suggests more work is needed to fully understand the effects of altitude, aircraft attitude, and airframe integration on the geolocation process. The simulated studies showed that flying closer to the target was the most reliable was to increase the accuracy of geolocation. It also suggested that if the physical system could be accurate to within the ±10 degrees that was anticipated from ground testing, the system could indeed produce reliable and accurate target position estimation provided that the UAV could fly both near the target and circle around it. The next chapter will summarize the findings of the work and provide steps forward to improve upon the work in the future.

V. Conclusions and Recommendations

5.1 Chapter Overview

This chapter provides a summary of the major results of the work, along with why it is significant and what further actions and research avenues can be taken to improve and advance the work.

5.2 Conclusions of Research

After studying the problem of low-cost UAV-based geolocation, a few conclusions were able to be drawn. The first was that performing an accurate geolocation of a ground emitter from a low-cost UAV platform is very difficult in practice. While ground testing of the direction finding system proved promising by being able to locate the emitter to within 20 meters, the flight test results were highly erratic and unable to replicate the results observed from ground tests. It was ultimately concluded that the set-up as tested was unable to produce accurate results. It was also concluded that no easy correction factor for the data existed based off of transformations done with the yawing and heading angles of the aircraft, along with roll sorting of the points.

On the simulation side, it was concluded that accurate geolocation of an emitter is possible with a high error sensor of ±10 degrees provided the UAV could fly sufficiently close to the target. A distance of up to 0.5 miles away showed good results, giving a position error ellipse radius of 0.13 miles and growing more accurate the closer the UAV moved towards the target. This result was observed as long as the UAV could circle the target to gain the additional angular information needed for accurate geolocation. It was

also observed that the accuracy of the results deteriorated as the flight path of the UAV was made further from the target location. Some of this degradation could be offset by adding in a filter to get rid of outlying data points. It was found that increasing the strictness of the filter did indeed decrease the size of the position error ellipse, but it did not always decrease the distance from the estimated target position to the true target position, meaning that the filter made the results more precise but not necessarily more accurate.

5.3 Significance of Research

The significance of this research mainly comes from the progress made on the physical testing side of the low-cost UAV geolocation problem. Although the system did not function as well as hoped, it still provides a good baseline for results and multiple paths forward for improvement in the future.

5.4 Recommendations for Future Research

Upon completion of this work, numerous opportunities for further work are evident. On the simulation side, more work can be done on the path planning side and implementing some control logic into the design, such that the UAV seeks to minimize the error in position location instead of just flying a preset path. An additional area of study could be to look at creating a line intersection code that mimics the directionality of the LOBs generated from the direction finder. Instead of treating each line as extending infinitely in both directions, the code would only extend the line in the direction towards the target. The effects that a code like this would have on the geolocation algorithm are currently unknown. Lastly, the effects of filtering the data in different ways, such as how

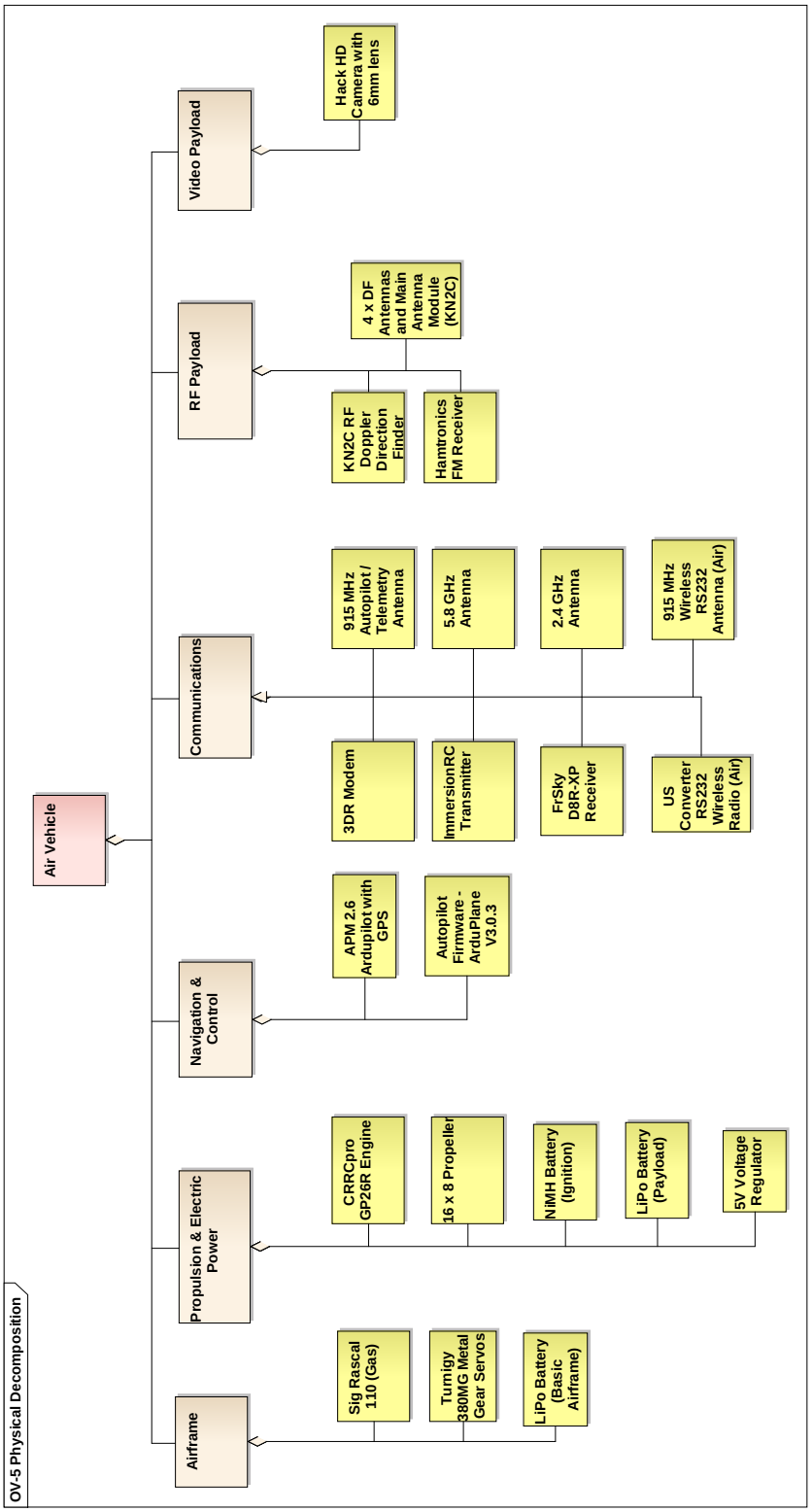
the baseline for comparison is constructed or how certain points are thrown out, could be investigated further.

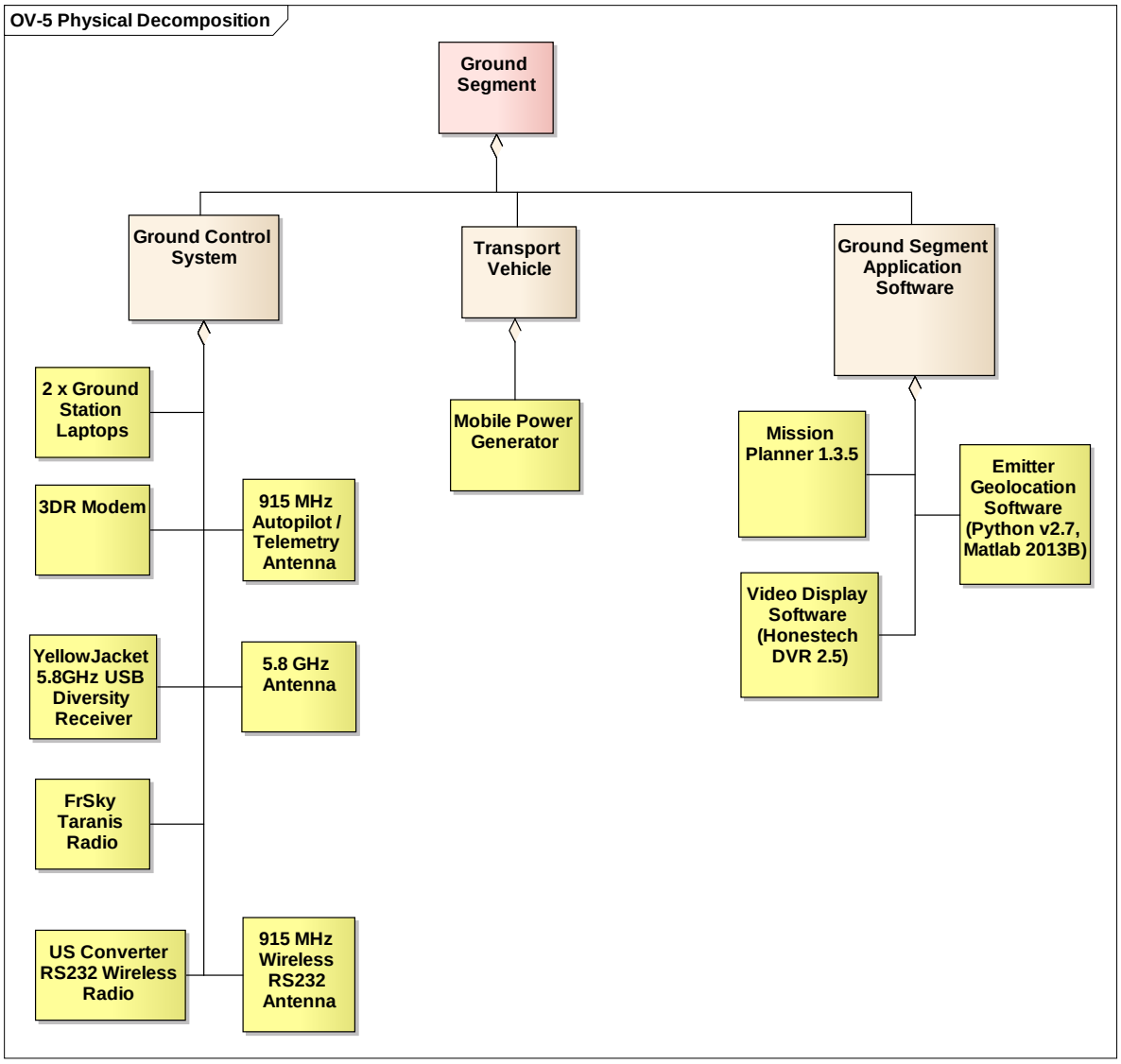
On the physical testing side, the first area for future study should be to investigate altitude and attitude effects on the geolocation results. This would require extending the current two-dimensional problem into a fully three-dimensional one. This could initially be tested on the ground by rotating the antenna array through a 90 degree sweep so that the antennas are no longer parallel to the target. By doing this in small increments and observing the resulting performance of the direction finding system, the effects of changing attitude could possibly be characterized. Special attention should be paid to the angles that were actually observed during flight test, which are shown in Figure 4.5. From there, it could be attempted to include aircraft altitude and orientation into the simulation scenarios to see if more accurate results are possible.

5.5 Summary

This research has investigated the problem of low-cost UAV-based geolocation of ground-based RF emitters from both the physical and theoretical sides. Although the flight test results were not as accurate as desired, valuable contributions have still been made and paths forward for future work have been discovered.

Appendix A – Physical Architecture




OV-5 Physical Decomposition
Ground Segment
Ground Control System
Transport Vehicle
Ground Segment Application Software
2 x Ground Station Laptops
3DR Modem
915 MHz Autopilot / Telemetry Antenna
YellowJacket 5.8GHz USB Diversity Receiver
5.8 GHz Antenna
FrSky Taranis Radio
US Converter RS232 Wireless Radio
915 MHz Wireless RS232 Antenna
Mobile Power Generator
Mission Planner 1.3.5
Emitter Geolocation Software (Python v2.7, Matlab 2013B)
Video Display Software (Honestech DVR 2.5)

Appendix B – Matlab Geolocation Code

```
% UAV GEOLOCATION SIMULATION
% Michael Magers

% This code was written to study the effects of path, uncertainty, and
% tolerance filter on UAV geolocation in partial fulfillment of Masters
% Thesis requirements at the Air Force Institute of Technology.

% To use the code, run the set-up section and then the section
% corresponding to the path you want to analyze. From that point there
% is two options: 1) Running the path and error ellipse visualization
% and 2)Running the Monte Carlo Simulations

% 1) To run the path and error ellipse visualization, lines 373 - 414
% should be commented out and the number of monte carlo simulations
% should be set to 1 in line 228.

% 2) To run the monte carlo simulations, the above lines should be made
% active again if they were commented out. Then lines 245-253, 258-268,
% 356, and 360 should be commented out. The number of monte carlo
% simulations also needs to be specified in line 228.

%  For each case the title of the plots created need to be changed to
% match the input conditions.

% The cone angle can be adjusted in line 42 and the filtering (sorting)
% can be adjusted in lines 318-333. For no sorting those lines should
% be commented out.

%% Set-Up Parameters of Simulation

clear all
close all
clc
format compact

% Input true target position (miles):
tx_true = 0;
ty_true = 0;

% Define uncertainty in bearing measurement in degrees(forms the cone
% lines)
uncert = 10;
sigma_beta=deg2rad(uncert);

% To change filter use line (319) or comment out lines (318-333) for no filter

% Set UAV velocity in MPH
vel = 35;
```

```
% Set sample rate time for measurements in seconds
t_sample = 30;


%% Flight Path - Straight Line Close
x_start = -2;                % Designate x starting position in miles
y_start = -.5;               % Designate y starting position in miles

ttotal  = 7; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector

heading_angle = 0; % Heading angle for UAV over path
x_dot         = vel*cos(heading_angle); % East velocity
y_dot         = vel*sin(heading_angle); % North velocity

y = y_start.*ones(length(t),1);     % North position is constant
x = zeros(length(t),1);             % Calculate east position
for a = 1:length(t)
    x(a) = x_start+x_dot*t(a);
end



%% Flight Path - Straight Line Far
x_start = -2;                % Designate x starting position in miles
y_start = -2;                % Designate y starting position in miles

ttotal  = 7; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector

heading_angle = 0; % Heading angle for UAV over path
x_dot         = vel*cos(heading_angle); % East velocity
y_dot         = vel*sin(heading_angle); % North velocity

y = y_start.*ones(length(t),1);     % North position is constant
x = zeros(length(t),1);             % Calculate east position
for a = 1:length(t)
    x(a) = x_start+x_dot*t(a);
end


%% Flight Path - Loiter Around Target Close
x_start = 0;
y_start = -.5;

ttotal  = 5.5; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector
```

```
dt      = thour/(points-1);

heading_angle = linspace(0,2*pi,points);
y(1) = y_start;
x(1) = x_start;
for a = 2:length(t)
    y(a) = vel*sin(heading_angle(a))*dt+y(a-1);
    x(a) = vel*cos(heading_angle(a))*dt+x(a-1);
end
x = x';
y = y';


%% Flight Path - Loiter Around Target Far
x_start = 0;
y_start = -2;

ttotal  = 20; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector
dt      = thour/(points-1);

heading_angle = linspace(0,2*pi,points);
y(1) = y_start;
x(1) = x_start;
for a = 2:length(t)
    y(a) = vel*sin(heading_angle(a))*dt+y(a-1);
    x(a) = vel*cos(heading_angle(a))*dt+x(a-1);
end
x = x';
y = y';


%% Flight Path - Loiter Outside Target Close
x_start = 0;
y_start = -1.5;

ttotal  = 5.5; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector
dt      = thour/(points-1);

heading_angle = linspace(0,2*pi,points);
y(1) = y_start;
x(1) = x_start;
for a = 2:length(t)
    y(a) = vel*sin(heading_angle(a))*dt+y(a-1);
    x(a) = vel*cos(heading_angle(a))*dt+x(a-1);
end
x = x';
y = y';
```

```
%% Flight Path - Loiter Outside Target Far
x_start = 0;
y_start = -6;

ttotal  = 20; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(t_sample/3600)); % Get # of observation points
t       = linspace(0,thour,points)';   % Generate time vector
dt      = thour/(points-1);

heading_angle = linspace(0,2*pi,points);
y(1) = y_start;
x(1) = x_start;
for a = 2:length(t)
    y(a) = vel*sin(heading_angle(a))*dt+y(a-1);
    x(a) = vel*cos(heading_angle(a))*dt+x(a-1);
end
x = x';
y = y';


%% Flight Path - Loiter Around Target Very Close
x_start = 0;
y_start = -.031;

ttotal  = .33; % Set total sim time in minutes
thour   = ttotal/60; % Convert to hours for sim
points  = round(thour/(2/3600)); % Get number of observation points
t       = linspace(0,thour,points)';   % Generate time vector
dt      = thour/(points-1);

heading_angle = linspace(0,2*pi,points);
y(1) = y_start;
x(1) = x_start;
for a = 2:length(t)
    y(a) = vel*sin(heading_angle(a))*dt+y(a-1);
    x(a) = vel*cos(heading_angle(a))*dt+x(a-1);
end
x = x';
y = y';


%% Flight Path - Button Hook
t = 0:.01:5;
tsamp = linspace(t(1),t(end),20)';
xpath = 1.*exp(-.8.*t).*sin(t);
ypath = -2.*exp(-.8.*t).*cos(t);
x(1) = xpath(1);
y(1) = ypath(1);
for a = 1:20
    x(a+1) = xpath(a*25);
    y(a+1) = ypath(a*25);
end
x = x';
```

```
y = y';
% hold on
% plot(x,y)
% scatter(x,y)
% axis([-1.5 1.5 -2.5 1])

tsim = zeros(length(x),1);
distance = 0;
for b = 1:length(x)-1;
    new = sqrt((x(b)-x(b+1))^2+(y(b)-y(b+1))^2);
    distance = distance + new;
    tsim(b+1) = distance/vel;
    distance;
end


%% Perform Geolocation
MCS = 100; % number of monte carlo simulations
mc1 = 1:MCS;

% Create uncertainty in the bearing measurement
error_meas  = deg2rad(0+1*randn(length(x),length(mc1)));
beta = zeros(length(x),length(mc1));
beta_err = zeros(length(x),2*length(mc1));

% Calculate bearing angles
for ii = 1:length(x)
    for iii = 1:length(mc1)
       beta(ii,iii) = atan2(ty_true-y(ii),tx_true-x(ii))+error_meas(ii,iii);  % Real
              angle + ERROR
         beta_err(ii,2*iii-1) = beta(ii,iii)-sigma_beta;
         beta_err(ii,2*iii)   = beta(ii,iii)+sigma_beta;
    end
end

% Start plotting the path and error ellipse results
figure(1)
hold on
grid on
xlabel('East Position (miles)')
ylabel('North Position (miles)')
title('Buttonhook w/ 5 deg uncertainty and 1 std filter')
scatter(tx_true,ty_true,30,'filled','g') %True target is here
line = 1.1*sqrt((x(1)-tx_true)^2+(y(1)-ty_true)^2); % Line magnitude
for plotting

% Perform the geolocation simulation for each monte carlo trial
for mc = 1:MCS
for len = 1:length(x);
    for k = 1:len
        % Plot UAV path
        plot(x(1:k),y(1:k),'g')

        % Plot UAV points where measurements are taken
```

```
            scatter(x(k),y(k),25,'filled','r')

         % Plot the cone lines at each point
          plot([x(k), x(k)+line*cos(beta_err(k,1))],[y(k), y(k)+line*sin(beta_err(k,1))],'r-')
          plot([x(k), x(k)+line*cos(beta_err(k,2))],[y(k), y(k)+line*sin(beta_err(k,2))],'r-')
        end

270     inter=[];
        [row,col]=size(inter);

    for m = 1:(len-1)
            for n = 1:(len-m)
                for n2 = 1:2

                 % Line intersections are calculated here through line
                 if  abs(tan(beta_err(m,2*mc-1))-tan(beta_err(len+1-n,n2))) < 10^(-2)
                 inter(row+1,1) = ((y(len+1-n,1)-y(m,1)) + (x(m,1)*tan(beta_err(m,2*mc))-
                       x(len+1-n,1)*tan(beta_err(len+1-n,n2))))/(tan(beta_err(m,2*mc))-
280                    tan(beta_err(len+1-n,n2)));
                 inter(row+1,2) = (inter(row+1,1)-x(m,1))*tan(beta_err(m,2*mc))+y(m,1);
                [row,col]=size(inter);
    %              plot(inter(row,1),inter(row,2), '.')


                elseif abs(tan(beta_err(m,2*mc))-tan(beta_err(len+1-n,n2))) < 10^(-2)
                inter(row+1,1) = ((y(len+1-n,1)-y(m,1)) + (x(m,1)*tan(beta_err(m,2*mc-1))-
                       x(len+1-n,1)*tan(beta_err(len+1-n,n2))))/(tan(beta_err(m,2*mc-1))-
                       tan(beta_err(len+1-n,n2)));
                inter(row+1,2) = (inter(row+1,1)-x(m,1))*tan(beta_err(m,2*mc-1))+y(m,1);
290             [row,col]=size(inter);
    %              plot(inter(row,1),inter(row,2), '.')

                else
                inter(row+1,1) = ((y(len+1-n,1)-y(m,1)) + (x(m,1)*tan(beta_err(m,2*mc-1))-
                       x(len+1-n,1)*tan(beta_err(len+1-n,n2))))/(tan(beta_err(m,2*mc-1))-
                       tan(beta_err(len+1-n,n2)));
                inter(row+1,2) = (inter(row+1,1)-x(m,1))*tan(beta_err(m,2*mc-1))+y(m,1);
                [row,col]=size(inter);
    %              plot(inter(row,1),inter(row,2), '.')

300             inter(row+1,1) = ((y(len+1-n,1)-y(m,1)) + (x(m,1)*tan(beta_err(m,2*mc))-
                       x(len+1-n,1)*tan(beta_err(len+1-n,n2))))/tan(beta_err(m,2*mc))-
                       tan(beta_err(len+1-n,n2)));
                 inter(row+1,2) = (inter(row+1,1)-x(m,1))*tan(beta_err(m,2*mc))+y(m,1);
                 [row,col]=size(inter);
    %               plot(inter(row,1),inter(row,2), '.')
                   end
                end
            end
        end

310     % Start estimating target position after 2 measurements
        if len >= 2
        avg = mean(inter);

        % Target is estimated to be average intersection position
        TGT(len-1,:) = avg;
```

```
        % Applying filtering (sorting) here
        if len > 5 % Choose when sorting starts
        filter = abs(1*std(inter)); % Input tolerance of sorting
320     for kk = 1:length(inter)
            if inter(kk,1) > avg(1,1)+filter(1,1)
                inter(kk,:) = 0;
            end
            if inter(kk,1) < avg(1,1)-filter(1,1)
                inter(kk,:) = 0;
            end
            if inter(kk,2) > avg(1,2)+filter(1,2)
                inter(kk,:) = 0;
            end
330         if inter(kk,2) < avg(1,2)-filter(1,2)
                inter(kk,:) = 0;
            end
        end

        % Get rid of points that fall outside of sorting tolerance
        con1 = inter(:,1) == 0;
        inter(con1,:) = []

        end
340
        % Plot intersection points and estimated target position
        plot(inter(:,1),inter(:,2),'.')
        scatter(avg(1,1),avg(1,2), 'r*')


        % Calculate error between intersection points and estimated position
        % for covariance matrix calculation
        for j = 1:length(inter(:,1))
            error(j,:)=[avg(1,1)-inter(j,1), avg(1,2)-inter(j,2)];
        end
350
        % Calculate covariance matrix, ellipse radius, and plot the error
        % ellipse
        C = cov(error);
        [eigvec,eigval]          = eig(C);
        longradius(len-1,1)      = sqrt(max(max(eigval)));
        h                        = error_ellipse(C,avg);

        % Pause here so you can advance through measurements one at a time and
        % observe error ellipse over time
360     pause
        end
    end


    % Store the results from one monte carlo run and then empty matrices for
    % next run
    storage(:,mc) = longradius;
    filter = [];
    error = [];
```

```
inter = [];
hold off
end

% Plot results from monte carlo tests here
figure(2)
hold on
xlabel('Time (min)')
ylabel('Max Ellipse Radius (miles)')
title('Loiter outside target close w/ 10 deg uncertainty and no filter')
% Plot results of each monte carlo test here
for p = 1:length(mc1)
    plot(t(2:length(storage(:,1))+1)*60,storage(:,p))
end
hold off

% Calculate average, min, and max monte carlo results here
for pp = 1:length(storage(:,1))
    avg_err(pp,1) = mean(storage(pp,:));
    max_err(pp,1) = max(storage(pp,:));
    min_err(pp,1) = min(storage(pp,:));
end

num_vec = t(2:length(storage(:,1))+1)*60;

% Plot average monte carlo result with error bars here
figure(3)
hold on
xlabel('Time (min)')
ylabel('Max Ellipse Radius (miles)')
title('Loiter Outside Target Close w/ 10 deg uncertainty and no filter')
errorbar(num_vec,avg_err,avg_err-min_err,max_err-avg_err)
hold off

% Get min ellipse radius in miles
[result1,I1] = min(avg_err);
result1

% Get estimated target position at that time and display absolute distance
% from the true target in meters
result2 = TGT(I1,:);
result2b = sqrt(result2(1,1)^2+result2(1,2)^2)*1609.34

% Get time at which error ellipse drops below .1 miles in minutes
first = find(avg_err < .1);
result3 = t(first(1)+1)*60
```

Appendix C - Flight Test Analysis Code

```
% Flight Test Data Analysis Code
% Michael Magers

% This code is used to analyze the flight test data collected from the UAV
% geolocation tests conducted at Camp Atterbury, IN. This analyzes the
% original data, data with correction factors, and data sorted by roll
% angle.

clear all
close all
clc

% Load in the flight data: FlightTest1 through FlightTest5
data2 = load('FlightTest1.mat');
data2 = struct2cell(data2);
data2 = cell2mat(data2);

% Read in the relevant data. The data in each column is given as 1) RF GPS
% time, 2) RF Latitude, 3) RF Longitude, 4) RF Heading, 5) RF Bearing, 6)
% Autopilot (AP) Latitude, 7) AP Longitude, 8) AP Bearing, 9) AP Roll, 10)
% AP Pitch, 11) AP Yaw, 12) AP Airspeed, 13) AP GPS Time, 14) AP Altitude.
% The time vectors appear to be corrupted or wrong though.

lat2    = data2(:,2); % Latitude
long2   = data2(:,3); % Longitude
rfhead2 = data2(:,4); % RF Heading
rfbear2 = data2(:,5); % RF Bearing
apbear2 = data2(:,8); % Autopilot Bearing
aproll2 = data2(:,9); % Autopilot Roll
apyaw2  = data2(:,11); % Autopilot Yaw
ref2    = [39.3453 -86.0121]; % Reference location for comparision

% Convert to relative meters for analysis
lat2m   = ((lat2-ref2(1,1)).*111030.25);
long2m  = ((long2-ref2(1,2)).*85675.27);

% Line magnitude for plotting
linemag = 50;

% Set up matrices for speed
lob2  = zeros(length(lat2),2);
lob2c = zeros(length(lat2),2);
PA2   = zeros(length(lat2),2);
PB2   = zeros(length(lat2),2);
PB2c   = zeros(length(lat2),2);


% Calculate the lines of bering , the lines of bearing with a correction
% factor, and the start and end points of the bearing lines
for i = 1:length(lat2)
    lob2(i,1) = linemag.*cosd(rfbear2(i))+long2m(i);
```

```
        lob2(i,2) = linemag.*sind(rfbear2(i))+lat2m(i);

        % Apply correction factor here (i.e. plus or minus yaw, roll, heading,
        % 45 or 90 degree offsets)
        lob2c(i,1) = linemag.*cosd(rfbear2(i)-apyaw2(i))+long2m(i);
        lob2c(i,2) = linemag.*sind(rfbear2(i)-apyaw2(i))+lat2m(i);

        % Start and end points of lines for line intersection code
        PA2(i,1) = long2m(i,1);
        PA2(i,2) = lat2m(i,1);
        PB2(i,1) = lob2(i,1);
        PB2(i,2) = lob2(i,2);
        PB2c(i,1) = lob2c(i,1);
        PB2c(i,2) = lob2c(i,2);
    end


    % Define sorting categories for the roll analysis: cat1 is less than -10
    % degrees of roll, cat2 is from -10 to 0, cat3 is from 0 to 10, and cat4 is
    % greater than 10 degrees roll
    cat1 = find(aproll2 < -10);
    cat2 = find(-10 <= aproll2 & aproll2 < 0);
    cat3 = find(0 <= aproll2 & aproll2 <= 10);
    cat4 = find(10 < aproll2);


    % Sort the data for the roll anaylsis, and plot results of the original,
    % correction factor, and roll sorted data
    figure(1)
    hold on
    grid on
    scatter(0,0,'*','r') % True target location at( 0,0)
    for j = cat1
        for jj = 1:length(j)
        scatter(long2m(j(jj)),lat2m(j(jj)))
        plot([long2m(j(jj),1) lob2(j(jj),1)],[lat2m(j(jj),1) lob2(j(jj),2)],'b--')
        end
    end
    for j = cat2
        for jj = 1:length(j)
        scatter(long2m(j(jj)),lat2m(j(jj)))
         plot([long2m(j(jj),1) lob2(j(jj),1)],[lat2m(j(jj),1) lob2(j(jj),2)],'r--')
        end
    end
    for j = cat3
        for jj = 1:length(j)
        scatter(long2m(j(jj)),lat2m(j(jj)))
        plot([long2m(j(jj),1) lob2(j(jj),1)],[lat2m(j(jj),1) lob2(j(jj),2)],'r')
        end
    end
    for j = cat4
        for jj = 1:length(j)
        scatter(long2m(j(jj)),lat2m(j(jj)))
        plot([long2m(j(jj),1) lob2(j(jj),1)],[lat2m(j(jj),1) lob2(j(jj),2)],'b')
        end
```

```
end

for i = 1:length(lat2)
     plot([long2m(i,1) lob2c(i,1)],[lat2m(i,1) lob2c(i,2)],'g')
end

tot = [cat2(:,1);cat3(:,1)];

% Use line intersection code to obtain least squares intersection point
% along with the errors from each line to that point
[int2,d2,e2]   = lineIntersect(PA2,PB2);
scatter(int2(1,1),int2(1,2),20,'r')

[int2c,d2c,e2c]   = lineIntersect(PA2,PB2c);
scatter(int2c(1,1),int2c(1,2),40,'r')

[int2r,d2r,e2r]   = lineIntersect(PA2(tot,:),PB2(tot,:));
scatter(int2r(1,1),int2r(1,2),40,'r')


% Calculate the covariance matrix from the errors and use the error ellispe
% code to plot the error ellipse. Orginal ellipse in blue, corrected in
% green, roll sorted in red.
error2 = [int2(:,1) - e2(:,1), int2(:,2)- e2(:,2)];
C2     = cov(error2);
h      = error_ellipse(C2,int2,'style','b');

error2c = [int2c(:,1) - e2c(:,1), int2c(:,2)- e2c(:,2)];
C2c     = cov(error2c);
hc      = error_ellipse(C2c,int2c,'style','g');

error2r = [int2r(:,1) - e2r(:,1), int2r(:,2)- e2r(:,2)];
C2r     = cov(error2r);
hr      = error_ellipse(C2r,int2r,'style','r');

xlabel('Relative meters')
ylabel('Relative meters')
title('Flight Test Data Analysis, true target at (0,0)')

% Length from estimated target position to true target position for
each
% case for comparison
dis1 = sqrt(int2(1,1)^2+int2(1,2)^2)
dis2 = sqrt(int2c(1,1)^2+int2c(1,2)^2)
dis3 = sqrt(int2r(1,1)^2+int2r(1,2)^2)
```

Appendix D - Line Intersection Code [40]

```
function [P_intersect,distances,error2P] = lineIntersect(PA,PB)
% Find intersection point of lines in 2D space, in the least squares sense.
% PA :           Nx2-matrix containing starting point of N lines
% PB :           Nx2-matrix containing end point of N lines
% P_Intersect : Best intersection point of the N lines, in least squares sense.
% distances   : Distances from intersection point to the input lines
% Anders Eikenes, 2012

Si = PB - PA; %N lines described as direction vectors
ni = Si ./ (sqrt(sum(Si.^2,2))*ones(1,2)); %Normalize direction vectors
nx = ni(:,1); ny = ni(:,2);
SXX = sum(nx.^2-1);
SYY = sum(ny.^2-1);
SXY = sum(nx.*ny);
S = [SXX SXY; SXY SYY ];
CX  = sum(PA(:,1).*(nx.^2-1) + PA(:,2).*(nx.*ny));
CY  = sum(PA(:,1).*(nx.*ny)  + PA(:,2).*(ny.^2-1));
C   = [CX;CY];
P_intersect = (S\C)';

if nargout>1
    N = size(PA,1);
    distances=zeros(N,1);
    error2P=zeros(N,2);

    for i=1:N %This is faster:
        ui=(P_intersect-PA(i,:))*Si(i,:)'/(Si(i,:)*Si(i,:)');
        distances(i)=norm(P_intersect-PA(i,:)-ui*Si(i,:));
        error2P(i,:)=P_intersect-PA(i,:)-ui*Si(i,:);
    end

    %for i=1:N %http://mathworld.wolfram.com/Point-LineDistance3-
Dimensional.html:
    %    distances(i) = norm(cross(P_intersect-PA(i,:),P_intersect-
PB(i,:))) / norm(Si(i,:));
    %end
end
end
```

Appendix E - Error Ellipse Calculation Code [41]

```
function h=error_ellipse(varargin)
% ERROR_ELLIPSE - plot an error ellipse, or ellipsoid, defining confidence region
%    ERROR_ELLIPSE(C22) - Given a 2x2 covariance matrix, plot the
%    associated error ellipse, at the origin. It returns a graphics handle
%    of the ellipse that was drawn.
%
%    ERROR_ELLIPSE(C33) - Given a 3x3 covariance matrix, plot the
%    associated error ellipsoid, at the origin, as well as its projections
%    onto the three axes. Returns a vector of 4 graphics handles, for the
%    three ellipses (in the X-Y, Y-Z, and Z-X planes, respectively) and for
%    the ellipsoid.
%
%    ERROR_ELLIPSE(C,MU) - Plot the ellipse, or ellipsoid, centered at MU,
%    a vector whose length should match that of C (which is 2x2 or 3x3).
%
%    ERROR_ELLIPSE(...,'Property1',Value1,'Name2',Value2,...) sets the
%    values of specified properties, including:
%      'C' - Alternate method of specifying the covariance matrix
%      'mu' - Alternate method of specifying the ellipse (-oid) center
%      'conf' - A value betwen 0 and 1 specifying the confidence interval.
%        the default is 0.5 which is the 50% error ellipse.
%      'scale' - Allow the plot the be scaled to difference units.
%      'style' - A plotting style used to format ellipses.
%      'clip' - specifies a clipping radius. Portions of the ellipse, -oid,
%        outside the radius will not be shown.
%
%    NOTES: C must be positive definite for this function to work properly.

default_properties = struct(...
  'C', [], ... % The covaraince matrix (required)
  'mu', [], ... % Center of ellipse (optional)
  'conf', 0.50, ... % Percent confidence/100
  'scale', 1, ... % Scale factor, e.g. 1e-3 to plot m as km
  'style', '', ...  % Plot style
  'clip', inf); % Clipping radius

if length(varargin) >= 1 & isnumeric(varargin{1})
  default_properties.C = varargin{1};
  varargin(1) = [];
end

if length(varargin) >= 1 & isnumeric(varargin{1})
  default_properties.mu = varargin{1};
  varargin(1) = [];
end

if length(varargin) >= 1 & isnumeric(varargin{1})
  default_properties.conf = varargin{1};
  varargin(1) = [];
end

if length(varargin) >= 1 & isnumeric(varargin{1})
  default_properties.scale = varargin{1};
```

```
  varargin(1) = [];
end

if length(varargin) >= 1 & ~ischar(varargin{1})
  error('Invalid parameter/value pair arguments.')
end

prop = getopt(default_properties, varargin{:});
C = prop.C;

if isempty(prop.mu)
  mu = zeros(length(C),1);
else
  mu = prop.mu;
end

conf = prop.conf;
scale = prop.scale;
style = prop.style;

if conf <= 0 | conf >= 1
  error('conf parameter must be in range 0 to 1, exclusive')
end

[r,c] = size(C);
if r ~= c | (r ~= 2 & r ~= 3)
  error(['Don''t know what to do with ',num2str(r),'x',num2str(c),'
matrix'])
end

x0=mu(1);
y0=mu(2);

% Compute quantile for the desired percentile
k = sqrt(qchisq(conf,r)); % r is the number of dimensions (degrees of
freedom)

hold_state = get(gca,'nextplot');

if r==3 & c==3
  z0=mu(3);

  % Make the matrix has positive eigenvalues - else it's not a valid
covariance matrix!
  if any(eig(C) <=0)
    error('The covariance matrix must be positive definite (it has non-
positive eigenvalues)')
  end

  % C is 3x3; extract the 2x2 matricies, and plot the associated error
  % ellipses. They are drawn in space, around the ellipsoid; it may be
  % preferable to draw them on the axes.
  Cxy = C(1:2,1:2);
```

```
  Cyz = C(2:3,2:3);
  Czx = C([3 1],[3 1]);

  [x,y,z] = getpoints(Cxy,prop.clip);
  h1=plot3(x0+k*x,y0+k*y,z0+k*z,prop.style);hold on
  [y,z,x] = getpoints(Cyz,prop.clip);
  h2=plot3(x0+k*x,y0+k*y,z0+k*z,prop.style);hold on
  [z,x,y] = getpoints(Czx,prop.clip);
  h3=plot3(x0+k*x,y0+k*y,z0+k*z,prop.style);hold on


  [eigvec,eigval] = eig(C);

  [X,Y,Z] = ellipsoid(0,0,0,1,1,1);
  XYZ = [X(:),Y(:),Z(:)]*sqrt(eigval)*eigvec';

  X(:) = scale*(k*XYZ(:,1)+x0);
  Y(:) = scale*(k*XYZ(:,2)+y0);
  Z(:) = scale*(k*XYZ(:,3)+z0);
  h4=surf(X,Y,Z);
  colormap gray
  alpha(0.3)
  camlight
  if nargout
    h=[h1 h2 h3 h4];
  end
elseif r==2 & c==2
  % Make the matrix has positive eigenvalues - else it's not a valid
covariance matrix!
  if any(eig(C) <=0)
    error('The covariance matrix must be positive definite (it has non-
positive eigenvalues)')
  end

  [x,y,z] = getpoints(C,prop.clip);
  h1=plot(scale*(x0+k*x),scale*(y0+k*y),prop.style);
  set(h1,'zdata',z+1)
  if nargout
    h=h1;
  end
else
  error('C (covaraince matrix) must be specified as a 2x2 or 3x3
matrix)')
end
%axis equal

set(gca,'nextplot',hold_state);

%-------------------------------------------------------------------
% getpoints - Generate x and y points that define an ellipse, given a 2x2
%   covariance matrix, C. z, if requested, is all zeros with same shape as
%   x and y.
```

```
function [x,y,z] = getpoints(C,clipping_radius)

n=100; % Number of points around ellipse
p=0:pi/n:2*pi; % angles around a circle

[eigvec,eigval] = eig(C); % Compute eigen-stuff
xy = [cos(p'),sin(p')] * sqrt(eigval) * eigvec'; % Transformation
x = xy(:,1);
y = xy(:,2);
z = zeros(size(x));

% Clip data to a bounding radius
if nargin >= 2
  r = sqrt(sum(xy.^2,2)); % Euclidian distance (distance from center)
  x(r > clipping_radius) = nan;
  y(r > clipping_radius) = nan;
  z(r > clipping_radius) = nan;
end

%--------------------------------------------------------------
function x=qchisq(P,n)
% QCHISQ(P,N) - quantile of the chi-square distribution.
if nargin<2
  n=1;
end

s0 = P==0;
s1 = P==1;
s = P>0 & P<1;
x = 0.5*ones(size(P));
x(s0) = -inf;
x(s1) = inf;
x(~(s0|s1|s))=nan;

for ii=1:14
  dx = -(pchisq(x(s),n)-P(s))./dchisq(x(s),n);
  x(s) = x(s)+dx;
  if all(abs(dx) < 1e-6)
    break;
  end
end

%--------------------------------------------------------------
function F=pchisq(x,n)
% PCHISQ(X,N) - Probability function of the chi-square distribution.
if nargin<2
  n=1;
end
F=zeros(size(x));

if rem(n,2) == 0
  s = x>0;
  k = 0;
  for jj = 0:n/2-1;
```

```
    k = k + (x(s)/2).^jj/factorial(jj);
  end
  F(s) = 1-exp(-x(s)/2).*k;
else
  for ii=1:numel(x)
    if x(ii) > 0
      F(ii) = quadl(@dchisq,0,x(ii),1e-6,0,n);
    else
      F(ii) = 0;
    end
  end
end

%-------------------------------------------------------------------
function f=dchisq(x,n)
% DCHISQ(X,N) - Density function of the chi-square distribution.
if nargin<2
  n=1;
end
f=zeros(size(x));
s = x>=0;
f(s) = x(s).^(n/2-1).*exp(-x(s)/2)./(2^(n/2)*gamma(n/2));

%-------------------------------------------------------------------
function properties = getopt(properties,varargin)
%GETOPT - Process paired optional arguments as 'prop1',val1,'prop2',val2,...
%
%   getopt(properties,varargin) returns a modified properties structure,
%   given an initial properties structure, and a list of paired arguments.
%   Each argumnet pair should be of the form property_name,val where
%   property_name is the name of one of the field in properties, and val is
%   the value to be assigned to that structure field.
%
%   No validation of the values is performed.
%
% EXAMPLE:
%   properties = struct('zoom',1.0,'aspect',1.0,'gamma',1.0,'file',[],'bg',[]);
%   properties = getopt(properties,'aspect',0.76,'file','mydata.dat')
% would return:
%   properties =
%         zoom: 1
%       aspect: 0.7600
%        gamma: 1
%         file: 'mydata.dat'
%           bg: []
%
% Typical usage in a function:
%   properties = getopt(properties,varargin{:})

% Process the properties (optional input arguments)
prop_names = fieldnames(properties);
TargetField = [];
for ii=1:length(varargin)
  arg = varargin{ii};
  if isempty(TargetField)
    if ~ischar(arg)
      error('Propery names must be character strings');
```

```
    end
    f = find(strcmp(prop_names, arg));
    if length(f) == 0
      error('%s ',['invalid property ''',arg,''''; must be one
of:'],prop_names{:});
    end
    TargetField = arg;
  else
    % properties.(TargetField) = arg; % Ver 6.5 and later only
    properties = setfield(properties, TargetField, arg); % Ver 6.1
friendly
    TargetField = '';
  end
end
if ~isempty(TargetField)
  error('Property names and values must be specified in pairs.');
end
```

Bibliography

[1]. A. Pages-Zamora, J. Vidal and D. H. Brooks, "Closed-form solution for positioning based on angle of arrival measurements," *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, 2002, pp. 1522-1526 vol.4.

[2]. T. S. Rappaport, J. H. Reed and B. D. Woerner, "Position location using wireless communications on highways of the future," in *IEEE Communications Magazine*, vol. 34, no. 10, pp. 33-41, Oct 1996.

[3]. Stansfield, R.G. "Statistical theory of D.F. fixing," in *Electrical Engineers - Part IIIA: Radiocommunication, Journal of the Institution of*, vol. 94, no. 15, pp. 762-770, March-April 1947.

[4]. Reza Zekavat; R. Michael Buehrer, "An Introduction to Direction-of-Arrival Estimation Techniques via Antenna Arrays," in *Handbook of Position Location:Theory, Practice and Advances* , 1, Wiley-IEEE Press, 2012, pp.279-317.

[5]. A. Amar, G. Leus and B. Friedlander, "Emitter position and velocity estimation given time and frequency differences of arrival," *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, Pacific Grove, CA, 2010, pp. 589-593.

[6]. A. Mikhalev and R. Ormondroyd, "Passive emitter geolocation using agent-based data fusion of AOA, TDOA and FDOA measurements," *Information Fusion, 2007 10th International Conference on*, Quebec, Que., 2007, pp. 1-6.

[7]. D. Musicki and W. Koch, "Geolocation using TDOA and FDOA measurements," *Information Fusion, 2008 11th International Conference on*, Cologne, 2008, pp. 1-8.

[8]. Reza Zekavat; R. Michael Buehrer, "Fundamentals of Time-of-Arrival-Based Position Locations," in *Handbook of Position Location:Theory, Practice and Advances* , 1, Wiley-IEEE Press, 2012, pp.175-212 .

[9]. D. R. Van Rheeden, B. C. Brown, J. C. Price, B. A. Abbott and G. C. Willden, "Automatic positioning of UAVs to optimize TDOA geolocation performance," *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, 2004, pp. 2.E.2-2.1-9 Vol.2.

[10]. N. Okello and D. Musicki, "Measurement Association for emitter geolocation with two UAVs," *Information Fusion, 2007 10th International Conference on*, Quebec, Que., 2007, pp. 1-8.

[11]. Paul Scerri, Robin Glinton, Sean Owens, David Scerri, and Katia Sycara. "Geolocation of RF Emitters by Many UAVs", AIAA Infotech@Aerospace 2007 Conference and Exhibit, Infotech@Aerospace Conferences, pp. 1-13.

[12]. Reza Zekavat; R. Michael Buehrer, "Fundamentals of Received Signal Strength-Based Position Location," in *Handbook of Position Location:Theory, Practice and Advances* , 1, Wiley-IEEE Press, 2012, pp.359-394.

[13]. Reza Zekavat; R. Michael Buehrer, "Source Localization: Algorithms and Analysis," in *Handbook of Position Location:Theory, Practice and Advances* , 1, Wiley-IEEE Press, 2012, pp.25-66.

[14]. MIT OpenCourseWare. “Chapter 3: Antennas.” http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-661-receivers-antennas-and-signals-spring-2003/readings/ch3new.pdf. February 2016.

[15]. “Bow Tie Antennas.” http://www.antenna-theory.com/antennas/wideband/bowtie.php. January 2016.

[16]. “Doppler Systems LLC Radio Direction Finders and Direction Finding Systems.” http://www.dopsys.com/principle/principle.html. January 2016.

[17]. “An FPGA Implementation Feasibility Study of the Correlative Interferometer Direction Finding Algorithm – Part 1: Algorithm Review.” http://www.nutaq.com/blog/fpga-implementation-feasibility-study-correlative-interferometer-direction-finding-algorithm-0. February 2016.

[18]. S. J. Fu, J. L. Vian and D. L. Grose, "Determination of ground emitter location," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 3, no. 12, pp. 15-18, Dec. 1988.

[19]. H. Witzgall, "A reliable Doppler-based solution for single sensor geolocation," *Aerospace Conference, 2013 IEEE*, Big Sky, MT, 2013, pp. 1-7.

[20]. Q. Liang and S. W. Samn, "UAV-based passive geolocation based on channel estimation," *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, Miami, FL, 2010, pp. 1821-1825.

[21]. Bamberger, Robert J., Moore, Jay G., Goonasekeram, Ravi P., Scheidt, David H. Autonomous Geolocation of RF Emitters Using Small, Unmanned Platforms. *Johns Hopkins APL Technical Digest,* Volume 32, Number 3, 2013, 636-646.

[22]. N. Okello and D. Musicki, "Emitter Geolocation with Two UAVs," *Information, Decision and Control, 2007. IDC '07,* Adelaide, Qld., 2007, pp. 254-259.

[23]. S. R. Semper and J. L. Crassidis, "Decentralized geolocation and optimal path planning using limited UAVs," *Information Fusion, 2009. FUSION '09. 12th International Conference on*, Seattle, WA, 2009, pp. 355-362.

[24]. D. J. Walter, K. Bryan, J. Stephens, C. Bullmaster and V. Chakravarthy, "Localization of RF emitters using compressed sensing with multiple cooperative sensors," *Aerospace and Electronics Conference (NAECON), 2012 IEEE National*, Dayton, OH, 2012, pp. 236-240.

[25]. Wang, Zhonghai, Chen, Genshe, Shen, Dan, Lin, Xingping, Blasch, Eric, Pham, Khanh. "A Low-Cost, Near-Real-Time Two-UAS-Based UWB Emitter Monitoring System." *IEEE AES Magazine November 2015.*

[26]. L. Marsh, D. Gossink, S. P. Drake and G. Calbert, "UAV Team Formation for Emitter Geolocation," *Information, Decision and Control, 2007. IDC '07,* Adelaide, Qld., 2007, pp. 176-181.

[27]. F. Fletcher, Branko Ristic and Darko Musicki, "Recursive estimation of emitter location using TDOA measurements from two UAVs," *Information Fusion, 2007 10th International Conference on*, Quebec, Que., 2007, pp. 1-8.

[28]. K. Doğançay, H. Hmam, S. Drake and A. Finn, "Centralized path planning for unmanned aerial vehicles with a heterogeneous mix of sensors," *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*, Melbourne, VIC, 2009, pp. 91-96.

[29]. F. Borrelli, D. Subramanian, A. U. Raghunathan and L. T. Biegler, "MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles," *American Control Conference, 2006*, Minneapolis, MN, 2006, pp. 5763-5768.

[30]. S. Hartzell, M. Haker, R. Martin, C. Taylor and A. Terzuoli, "AOA geolocation for fast-movers using nonlinear optimization," *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, Quebec City, QC, 2014, pp. 1855-1858.

[31]. Y. Zhang and J. Gao, "In-flight route re-planning for endurance reconnaissance unmanned aerial vehicles," *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, Shenzhen, 2008, pp. 1-5.

[32]. G. Wang, Q. Li and L. Guo, "Multiple UAVs Routes Planning Based on Particle Swarm Optimization Algorithm," *Information Engineering and Electronic Commerce (IEEC), 2010 2nd International Symposium on*, Ternopil, 2010, pp. 1-5.

[33]. "5 CH BlitzRCWorks Super Sky Surfer RC Sailplane Glider." http://www.bananahobby.com/5-ch-blitzrcworks-super-sky-surfer-rc-sailplane-glider-arf.html. October 2015.

[34]. "APM 2.6." https://3dr.com/kb/apm-2-6/. October 2015.

[35]. "The HANDI-Finder." http://www.handi-finder.com/. December 2015.

[36]. "Doppler Direction Finder." http://www.allspectrum.com/ramsey/DDF1/DDF1.pdf. March 2016.

[37]. "Radio Direction Finder DF2020T (Kit)." http://kn2c.us/. October 2015.

[38]. Oborne, Michael. "Mission Planner Home." http://planner.ardupilot.com/. March 2015.

[39]. Vermeer, Martin. "Statistical uncertainty and error propagation." http://users.aalto.fi/~mvermeer/uncertainty.pdf. March 27, 2014.

[40]. Eikenes, Anders. "Intersection point of lines in 2D space." http://www.mathworks.com/matlabcentral/fileexchange/37192-intersection-point-of-lines-in-3d-space. 2012.

[41]. Johnson, AJ. "error_ellipse." http://www.mathworks.com/matlabcentral/fileexchange/4705-error-ellipse. April 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
01-10-2013	Master's Thesis	October 2013 – March 2016

TITLE AND SUBTITLE

Geolocation of RF Emitters Using a Low-Cost UAV-Based Approach

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Magers, Michael A., Civilian, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-16-M-258

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Lifecycle Management Center, Flight Mechanics Branch
Building 28, Monahan Way, WPAFB, OH
ATTN: Jack Browne
John.browne.1@us.af.mil
(937) 255-8063

10. SPONSOR/MONITOR'S ACRONYM(S)

AFLCMC/EZFT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The proliferation of unmanned aerial vehicles (UAVs) in both military and civilian settings has prompted great interest in finding new and innovative ways to utilize these tools. One such application is to locate ground-based radio emitters from a UAV platform. The goal of this research is to study the feasibility of a low-cost (on the order of $1000) UAV geolocation platform. To accomplish this goal, a series of both real-world flight testing and computer simulated scenarios were conducted. Simulations for different sensor uncertainties and approach path scenarios such as loiter and button hook patterns were investigated. Results showed that a high uncertainty sensor of ±10 degrees was able to reliably geolocate the target provided it could fly sufficiently close to the emitter location. For the physical testing, a commercial-off-the-shelf Doppler direction finding unit was chosen as the method of performing the geolocation. Ground testing proved promising, locating the emitter to within 20 meters. However, flight testing showed poor results and was unable to locate the target. Areas of future work that could improve upon these results include investigating how altitude and antenna orientation variations caused by the movement of the aircraft affect the performance of the direction finding unit.

15. SUBJECT TERMS
Geolocation, RF, UAV

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Richard Cobb, AFIT/ENY
U	U	U	UU	112	19b. TELEPHONE NUMBER *(Include area code)* (937)255-3636 x4559 Richard.Cobb@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18